

Impact of Clustering on the Performance of Network De-anonymization

C.F. Chiasserini, M. Garetto, E. Leonardi

Abstract—Recently, graph matching algorithms have been successfully applied to the problem of network de-anonymization, in which nodes (users) participating to more than one social network are identified only by means of the structure of their links to other members. This procedure exploits an initial set of seed nodes large enough to trigger a percolation process which correctly matches almost all other nodes across the different social networks. Our main contribution is to show the crucial role played by clustering, which is a ubiquitous feature of realistic social network graphs (and many other systems). Clustering has both the effect of making matching algorithms more vulnerable to errors, and the potential to dramatically reduce the number of seeds needed to trigger percolation, thanks to a wave-like propagation effect. We demonstrate these facts by considering a fairly general class of random geometric graphs with variable clustering level, and showing how clever algorithms can achieve surprisingly good performance while containing matching errors.

I. INTRODUCTION

The advent of online social networks, and their massive worldwide penetration, can be well considered as one of the most influential changes brought by information and communication technologies into our lives during the last decade, with profound impact on all aspects of economy, society and culture. The extraordinary capitalization of the companies running these (typically free) online services can be explained by the huge amount of valuable information that can be extracted from the traces of activities performed by billions of users. Such information allows, for example, to build user profiles that can be effectively used for targeted advertisements, marketing and social surveys, and many other profitable business run by service providers and third parties. Privacy concerns raised by the collection, analysis and distribution of personal data, exposed more or less consciously by active users, have been recently hotly debated in the media. User privacy is especially threatened when data collected from different systems is combined together to construct richer and more accurate user profiles.

In this work we are specifically concerned with the problem of identifying users participating to different online social networks¹. We emphasize that this problem can be perceived by people in totally different ways. Some users would prefer to hide any Personal Identifiable Information (PII) while using a service, and they see any attempt to correlate accounts created

C.F. Chiasserini and E. Leonardi are with Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy, email: surname@polito.it; M. Garetto is with Dipartimento di Informatica, Università di Torino, Italy email: michele.garetto@unito.it

¹More in general, we are interested in any sort of communication system assigning some kind of (unique) ID to users, typically as a result of a new registration/account creation (including traditional communication services such as email and cellular networks).

in different systems as a severe violation of their privacy. Other users instead are more than happy to merge or link together their various accounts, as this turns out to be convenient to the user itself. For example, ‘social logins’ allow users to use existing accounts on social networks to directly sign into other services (different applications, websites, public Wi-Fi hotspots).

In our work, we are specifically interested in privacy issues, and consider the case of an ‘attacker’ trying to identify users belonging to two different social networks (without their consent). Recently, security experts have made the dramatic discovery that user privacy cannot be guaranteed when traces of communication activities are made available after applying the simple anonymization procedure which replaces real ID’s by random labels [1].

A standard way to formalize the user identification problem is the following: each communication system (e.g., a given social network) generates (from the traces of user activities) a ‘contact graph’ in which nodes represent anonymized users, and edges denote who has come in contact with whom. The attacker then runs a *graph matching* algorithm on the contact graphs generated by different systems, which in the hardest case can make use only of the topologies of these graphs, without any additional side information [2]. The majority of algorithms proposed so far to achieve this goal are facilitated by an initial set of already matched nodes (called seeds). This is actually a realistic case, since, as explained above, some users explicitly link their accounts in different systems ‘for free’. Many proposed matching strategies, based on heuristic algorithms, work by progressively expanding the set of already matched nodes, trying to identify all of the other nodes [1], [3], [4]. In particular, in their seminal paper Narayanan and Shmatikov [1] were able to identify a large fraction of users having account on both Twitter and Flickr (with only 12% error ratio).

Significant progress has also been made towards theoretical understanding of the feasibility of network de-anonymization (in the first place), and of the asymptotic performance of graph matching algorithms applied to large systems. Recent analytical work has adopted the following convenient probabilistic generation model for two contact graphs \mathcal{G}_1 and \mathcal{G}_2 : we consider the (inaccessible) ‘ground-truth’ graph \mathcal{G}_T representing true social relationships among people, and then assume that \mathcal{G}_1 is obtained by independently sampling each edge of \mathcal{G}_T with probability s (similarly, and independently, \mathcal{G}_2). Specifically, when the social network \mathcal{G}_T is modeled as an Erdős–Rényi random graph, it has been shown in [5] that, under mild conditions, users participating in two different social networks can be successfully matched by an attacker

with unlimited computation power, even without seeds. In the case of Erdős–Rényi random graphs, in [6] authors have also proposed a practical identification algorithm based on bootstrap percolation [7] and they have shown an interesting phase transition phenomenon in the number of seeds that are required for network de-anonymization. The results in [6] have been recently extended to the more realistic case in which contact graphs are scale-free (power law) random graphs. In particular, by modeling them as Chung–Lu graphs, [8] and [9] have independently shown that a much smaller set of seeds is sufficient to trigger the percolation-based matching process originally studied in Erdős–Rényi graphs.

While previous work has captured the impact of power-law degree distribution on percolation graph matching, another essential feature of real social networks, namely, clustering, has not been investigated so far. Interestingly, in [6] authors attempted to apply their basic algorithm also to highly clustered random geometric graphs, observing almost total failure (error rates above 50%). This preliminary finding has been the starting point of our work. In this paper we consider a fairly general model of random geometric graphs that allows us to incorporate various levels of clustering in contact graph, without concurrently generating a scale-free structure. By so doing, we separate the (unkown) impact of clustering from the (known) impact of power law degree, going back to the original case of Erdős–Rényi graphs and exploring a totally different, ‘orthogonal’ direction. Our main findings are as follows:

(i) Clustered networks can be indeed largely prone to matching errors when we naively apply the method proposed in [6]. Such errors can be mitigated and asymptotically eliminated by an improved matching algorithm still based on bootstrap percolation;

(ii) Once errors are eliminated, clustering turns out to have a surprising beneficial effect on the performance of graph matching, thanks to a wave-like propagation phenomenon that allows to progressively identify all nodes starting from a very small, *compact* set of seeds;

(iii) In contrast with previous results derived for Erdős–Rényi and Chung–Lu graphs in [6], [8], we show that the minimum number of seeds required for network de-anonymization increases as the average node degree of the graph grows.

Our results are qualitatively validated via experiments with real social network graphs. We emphasize that, although we focus on network de-anonymization, we do not cast our results exclusively to this problem. Indeed, the results we derive have much broader applicability since graph matching is a general problem arising in many different domains, ranging from computer graphics to bioinformatics.

II. NOTATION AND PRELIMINARIES

Without loss of generality, we assume that $\mathcal{G}_T(\mathcal{V}, \mathcal{E})$, $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ have the same set of nodes (or vertices) with cardinality n , i.e., $\mathcal{V}_1 = \mathcal{V}_2 = \mathcal{V}$ ². Similarly to previous work [4], [5], [6], [8], [9] we assume that edges

²This assumption can be easily removed by considering that only the intersection of vertices belonging to \mathcal{G}_1 and \mathcal{G}_2 has to be de-anonymized.

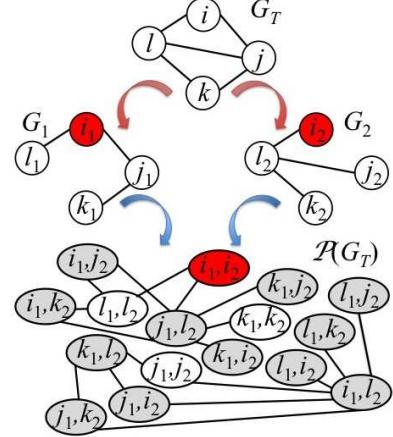


Fig. 1. An example of \mathcal{G}_1 and \mathcal{G}_2 obtained from \mathcal{G}_T by independent edge sampling, and of the pairs graph $\mathcal{P}(\mathcal{G}_T)$. Seeds are highlighted in red. In $\mathcal{P}(\mathcal{G}_T)$, good pairs are highlighted in white and bad pairs in grey.

in \mathcal{G}_1 and \mathcal{G}_2 are obtained by independently sampling each edge of \mathcal{G}_T with probability s . Specifically, each edge in \mathcal{G}_T is assumed to be (independently) sampled twice, the first time to determine its presence in \mathcal{E}_1 , the second time to determine its presence in \mathcal{E}_2 . This model is a reasonable approximation of real systems which permits obtaining fundamental analytical insights.

To match \mathcal{G}_1 and \mathcal{G}_2 , we build the pairs graph $\mathcal{P}(\mathcal{V}, \mathcal{E})$, with $\mathcal{V} \subseteq \mathcal{V}_1 \times \mathcal{V}_2$ and $\mathcal{E} \subseteq \mathcal{E}_1 \times \mathcal{E}_2$. In $\mathcal{P}(\mathcal{V}, \mathcal{E})$ there exists an edge between $[i_1, j_2]$ and $[k_1, l_2]$ iff edge $(i_1, k_1) \in \mathcal{E}_1$ and edge $(j_2, l_2) \in \mathcal{E}_2$. We will slightly abuse the notation and denote the pair graph associated to a generic ground-truth graph \mathcal{G}_T simply as $\mathcal{P}(\mathcal{G}_T)$. Fig. 1 shows the pairs graph built from a toy example.

We will refer to pairs $[i_1, i_2] \in \mathcal{P}(\mathcal{G}_T)$, whose vertices correspond to the same vertex $i \in \mathcal{G}_T$, as good pairs, and to all others (e.g., $[i_1, j_2]$) as bad pairs. Also, we will refer to two pairs such as $[i_1, j_2]$ and $[i_1, l_2]$, or $[i_1, j_2]$ and $[k_1, j_2]$, as conflicting. Finally, two adjacent pairs on $\mathcal{P}(\mathcal{G}_T)$ will be referred to as neighbors. The seed set³ will be denoted by $\mathcal{A}_0(n) \subset \mathcal{V}$, with cardinality a_0 .

We now briefly describe the Percolation Graph Matching (PGM) algorithm originally proposed in [6]. The PGM algorithm maintains an integer counter (initialized to zero) for any pair of $\mathcal{P}(\mathcal{G}_T)$ that may still be matched. It exploits a set \mathcal{A}_t , indexed by time step t , which is initialized (for $t = 0$) with the seed pairs. At any given time $t \geq 0$, the PGM algorithm extracts at random one pair from \mathcal{A}_t matching it, and increases by one the counter associated to each of its neighbor pair in $\mathcal{P}(\mathcal{G}_T)$. Then the algorithm adds to \mathcal{A}_{t+1} all pairs whose counter has reached r at time t with the exception of those pairs that are in conflict with either any of the already matched pairs or any of the pairs in \mathcal{A}_t . The algorithms stops when $\mathcal{A}_t = \emptyset$. It is straightforward to see that PGM takes at most n steps to terminate.

In the case where \mathcal{G}_T is an Erdős–Rényi random graph,

³We will refer to the seed set as a subset of vertices, or, equivalently, of good vertex pairs, that have been identified a-priori.

previous work [6] has established the following lower bound on the number of seeds that are needed to correctly match almost all nodes without errors.

Critical seed set size for Erdős-Rényi graphs [6]. Let \mathcal{G}_T be an Erdős-Rényi random graph $G(m, p)$. Let $r \geq 4$. Denote by a_c the critical seed set size:

$$a_c = \left(1 - \frac{1}{r}\right) \left(\frac{(r-1)!}{m(ps^2)^r}\right)^{\frac{1}{r-1}}. \quad (1)$$

For $m^{-1} \ll ps^2 \leq s^2 m^{-\frac{3.5}{r}}$, we have that, if $a_o/a_c \rightarrow a > 1$, the PGM algorithm matches w.h.p. a number of good pairs equal to $m - o(m)$ (i.e., all vertex pairs except for a negligible fraction) with no errors.

Critical seed set size for random graphs bounded by Erdős-Rényi graphs. Let $\mathcal{H}(\mathcal{V}, \mathcal{E}_H)$ and $\mathcal{K}(\mathcal{V}, \mathcal{E}_K)$ be two random graphs insisting on the same set of vertices \mathcal{V} , where $\mathcal{E}_H \subseteq \mathcal{E}_K$, i.e., \mathcal{E}_H can be obtained by sampling \mathcal{E}_K . We define the following partial order relationship: $\mathcal{H}(\mathcal{V}, \mathcal{E}_H) \leq_{st} \mathcal{K}(\mathcal{V}, \mathcal{E}_K)$. Given that, below we extend our result in [8].

Theorem 1: Consider \mathcal{G}_T satisfying: $G(m, p_{\min}) \leq_{st} \mathcal{G}_T \leq_{st} G(m, p_{\max})$ with $p_{\min} \leq p_{\max}$. Applying the PGM algorithm to $\mathcal{P}(\mathcal{G}_T)$ guarantees that $m - o(m)$ good pairs are matched with no errors w.h.p., provided that:

1. $m \rightarrow \infty$;
2. $p_{\min} = \Theta(p_{\max})$ and $p_{\min} \gg m^{-1}$;
3. $p_{\max} \leq m^{-\frac{3.5}{r}}$;
4. $\lim_{m \rightarrow \infty} a_o/a_c > 1$, with a_c computed from (1) by setting $p = p_{\min}$.

Also, under conditions 1)-4), the PGM successfully matches w.h.p. all the correct pairs (with no errors) also in any subgraph \mathcal{G}'_T of \mathcal{G}_T that comprises a finite fraction of vertices of \mathcal{G}_T and all the edges between the selected vertices. The proof can be found in Appendix A

Corollary 1: Under the same conditions as in Theorem 1, the PGM algorithm can be successfully applied to an imperfect pairs graph $\hat{\mathcal{P}}_x \subset \mathcal{P}(\mathcal{G}_T)$ comprising a finite fraction of the pairs in $\mathcal{P}(\mathcal{G}_T)$ and satisfying the following constraint: a bad pair $[i_1, j_2] \in \mathcal{P}(\mathcal{G}_T)$ is included in $\hat{\mathcal{P}}(\mathcal{G}_T)$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}(\mathcal{G}_T)$.

Under the above conditions, the objective of this work is to design and analyze the network de-anonymization process when the ground-truth graph, \mathcal{G}_T , exhibits different levels of nodes clustering. In particular, given \mathcal{G}_T , \mathcal{G}_1 and \mathcal{G}_2 , we aim to determine the minimum size of the seed set that is required to successfully identify w.h.p. all good vertex pairs in $\mathcal{P}(\mathcal{G}_T)$ with no errors. To this end, due to the big size of social network graphs, we perform an asymptotic analysis, i.e., we consider the number of vertices in \mathcal{G}_T to grow very large ($n \rightarrow \infty$).

III. CLUSTERED NETWORK MODEL

As detailed below, we model the social graph \mathcal{G}_T as a geometric random graph. At the end of this section, we highlight how our model well captures node clustering and how it can represent network graphs with different values of clustering coefficient.

We assume that nodes are located in a k -dimensional space corresponding to the hyper-cube⁴ $\mathcal{H} = [0, 1]^k \subset \mathbb{R}^k$, where the k dimensions correspond to different attributes of the user nodes. We consider the nodes to be independently and uniformly distributed over \mathcal{H} . Given any two vertices $i, j \in \mathcal{V}$, with $i \neq j$, edge (i, j) exists in the graph with probability p_{ij} that depends on the Euclidean distance d_{ij} between the respective positions of the two vertices in \mathcal{H} . We consider the following generic law for p_{ij} :

$$p_{ij} = K(n)f(d_{ij}). \quad (2)$$

In (2), f is a non-increasing function of the distance, and $K(n)$ is a normalization constant introduced to impose a desired average node degree, $D(n)$, which is assumed to be the same for all nodes. It is customary in random graph models representing realistic systems to assume that the average node degree is not constant, but it increases with n due to network densification. Also, although a common choice is to assume $D(n) = \Theta(\log n)$, in our model we consider $D(n) = \Omega(\log n)$ so as to encompass almost all systems of practical interest.

Since we are interested in the asymptotic performance of graph de-anonymization as n grows large, it is convenient to further characterize the shape of function f as follows. Let us define $C(n)$ to be at least equal to the minimal (in order sense) distance between nodes in \mathcal{H} , i.e., $n^{-1/k}$. We assume that $f(d)$ is equal to 1 for all distances $0 < d < C(n)$. This implies that $K(n)$ must be less than or equal to 1 to obtain a proper probability function. For distances larger than $C(n)$, we assume that f decays according to a power-law with exponent β , with $\beta > 0$. In summary,

$$f(d_{ij}) = \min \left\{ 1, \left(\frac{C(n)}{d_{ij}} \right)^\beta \right\}. \quad (3)$$

The above characterization of the shape of $f(d)$ is fairly general and allows accounting for different levels of node clustering. In particular, our random-graph model degenerates into a standard Erdős-Rényi graph when $C(n) = \Theta(1)$, with arbitrary β . For $\beta \rightarrow \infty$, instead, we have a geometric graph, i.e., edges can be established only between nodes whose distance is smaller than or equal to $C(n)$.

The average node degree is:

$$D(n) = \Theta \left(n K(n) \left(C^k(n) + C^\beta(n) \int_{C(n)}^1 \rho^{k-1-\beta} d\rho \right) \right). \quad (4)$$

Now, from (4) it follows that for $\beta > k$ the dominant component of the neighbors of a given node lie at a distance $\Theta(C(n))$ from it, while for $\beta < k$ only a marginal fraction of the neighbors of a node lie at distance $o(1)$ from it. Since we are interested in graphs with significant node clustering (so as to mimic real-world social networks), we restrict our analysis to the case $\beta > k$. In this case, the average node degree is given by:

$$D(n) = \Theta(n K(n) C^k(n)). \quad (5)$$

⁴To avoid border effects, we assume wrap-around conditions (i.e., a torus topology).

Because by construction $K(n) \leq 1$, the average node degree is constrained to be $O(nC^k(n))$. Moreover, given that we assume $D(n) = \Omega(\log n)$, we have $C(n) = \Omega\left(\left(\frac{\log n}{n}\right)^{\frac{1}{k}}\right)$.

The clustering coefficient turns out to be $\Theta(K(n))$, as a direct consequence of the fact that the major part of the neighbors of a node lie at a distance $\Theta(C(n))$ from it. In the following, we will slightly abuse the language and refer to groups of vertices that lie in sub-regions of side $\Theta(C(n))$ as *clusters*. Furthermore, we observe that, given the above expressions, the ratio of the clustering coefficient ($\Theta(K(n))$) to the graph density⁵ ($\Theta(D(n)/n)$) is $\Theta(1/C^k(n))$. This implies that our graph exhibits a high level of clustering. Indeed, since in general $C^k(n) = o(1)$, the probability that two nodes are connected conditioned on the fact that they have a common neighbor, is higher (in order sense) than the average probability that any two nodes are connected. It follows that, $K(n)$ and $C(n)$ result to be the key model parameters through which we can directly control the clustering coefficient of the graph as well as the graph density. Thus, they play a crucial role in the analysis we present below.

IV. OVERVIEW AND MAIN RESULTS

In our analysis we address two cases: clusters with relatively sparse structure, i.e., $K(n) = o((nC^k(n))^{-\gamma})$ for some $\gamma > 0$, and clusters with extremely dense (up to a quasi-clique) structure, i.e., $K(n) = \omega((nC^k(n))^{-\gamma})$ for any $\gamma > 0$.

In the former case the cluster density goes to zero sufficiently fast as the number of nodes within the cluster goes to infinity ($nC^k(n) \rightarrow \infty$). On the contrary, the latter corresponds to a cluster density that either is bounded away from zero or goes to zero very slowly, with $K(n) = \Theta(1)$ being a particularly relevant sub-case.

We observe that, in the case of relatively sparse cluster structure, the density of edges between nodes within a cluster is not excessively large and, thus, PGM can be safely applied without the risk of incurring in significant matching errors. We therefore apply the following procedure to determine the minimum set size required for successful graph de-anonymization. We assume that the set of seeds lie in a small sub-region of \mathcal{H} of size $\Theta(C(n))$ (i.e., within a cluster). Then, through the PGM algorithm, we de-anonymize all nodes that lie sufficiently close (within a prefixed distance) from the seeds. Once a significant bulk of pairs has been matched in this sub-region, the de-anonymization procedure is performed by successfully matching, at every stage, pairs that are sufficiently close to the previously matched pairs. Note that, starting from the second stage on, we do not apply PGM any longer but a simpler proximity-based strategy, matching those pairs that have a sufficiently large number of neighbors among the pairs matched at earlier stages. The way the matching procedure evolves is exemplified in Fig. 2.

In the case of dense cluster structure, the whole procedure is slightly more complex in light of the fact that the clustering

⁵Given a generic graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the graph density is defined as $\frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}|-1)}$. It can be interpreted as the probability that an edge exists between two randomly selected nodes of the graph.

TABLE I
MAIN RESULTS

Scenario	Minimum seed size
$K(n) = \omega((nC^k(n))^{-\gamma}), \forall \gamma > 0$	$O((nC^k(n))^{\epsilon}) \forall \epsilon > 0$
$K(n) = o((nC^k(n))^{-\gamma}), \text{with } \gamma > 0$	$\Theta\left(\frac{\log n C^k(n)}{K(n)}\right)$

coefficient is larger, thus considering short edges while running the PGM algorithm would lead to matching a large number of bad pairs (as their counters will likely exceed the threshold r). It follows that we have to ignore all edges whose length is too short (shorter than a properly defined threshold $\omega(C(n))$), in order to guarantee that almost no errors are made. More specifically, first we consider two groups of nodes that reside in two sub-regions of \mathcal{H} of side $h(n) = \Theta(C(n))$, which are taken sufficiently apart one from the other (see Fig. 3). Again, we assume that an opportune number of seeds is included in each sub-region. To de-anonymize all nodes in the sub-regions, we modify the PGM algorithm so that only the edges between the two different sub-regions are exploited. Then, by leveraging the presence of dense clusters, we show that, given two nodes in \mathcal{H} , their mutual distance can be estimated quite precisely. Thus, given a sub-region where nodes have already been matched, we can select a set of nodes that are again sufficiently apart from the others and repeat the above procedure. The procedure can be iterated till almost all good pairs are successfully matched.

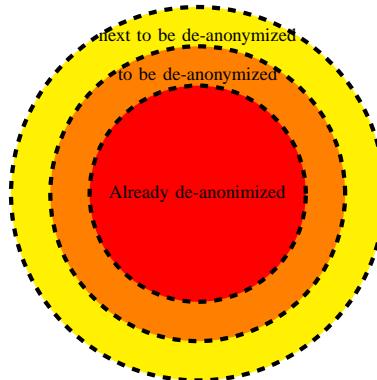


Fig. 2. Graphical representation of the de-anonymization procedure for $K(n) = o((nC^k(n))^{-\gamma})$.

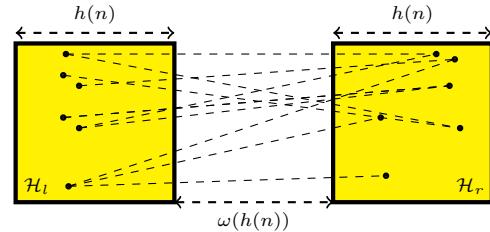


Fig. 3. Graphical representation of bipartite graph construction for $K(n) = \omega((nC^k(n))^{-\gamma})$.

In Table IV, we summarize our results on the minimum size of the seed set that is required for successful network

de-anonymization, when seeds are taken from compact sub-regions in \mathcal{H} . Observe that the minimum number of seeds depends on both $K(n)$ and $C(n)$ while it is independent of β . Specifically, in the regime of dense cluster structure (first raw of the table), the minimum number of seeds can be simply expressed in terms of the average number of nodes falling within a cluster ($nC^k(n)$). Indeed, a seed set whose size is equal to $(nC^k(n))^\epsilon$, for some ϵ , is enough to guarantee an almost complete successful network de-anonymization. In the relevant case in which $C(n) = \Theta(\frac{\log n}{n})$ (i.e., when the average degree of the graph $D(n) = \Theta(\log n)$), the above expression degenerates into $(\log n)^\epsilon$. This last expression permits grasping immediately the potential impact of node clustering on de-anonymization techniques. Furthermore, somehow surprisingly, the minimum seed set size increases when we increase the average degree of the graph nodes, by increasing $C(n)$. We remark that this is in sharp contrast with previous results derived for Erdős–Rényi and Chung–Lu graphs in [6], [8]. The intuition behind this result is that, by increasing $C(n)$, we increase also the cluster size making the problem of identifying nodes (users) within a cluster intrinsically more challenging. At last, when clusters become sparser (second raw of the table), de-anonymization techniques become less effective, and the minimum seed set size shows inverse proportional dependency on $K(n)$.

V. SPARSE CLUSTERS

In this case, we assume $K(n) = o((nC^k(n))^{-\gamma})$, for some $\gamma > 0$, and a set of seeds \mathcal{A}_0 ($|\mathcal{A}_0| = a_0$) whose maximum mutual distance is $d_s = O(C(n))$.

As the first step, we show how nodes in \mathcal{H} that lie sufficiently close to seeds can be identified. To this end, we start by defining two sub-regions, $\mathcal{H}_{\text{in}} \subset \mathcal{H}$ and $\mathcal{H}_{\text{out}} \subset \mathcal{H}$. Intuitively, \mathcal{H}_{in} (\mathcal{H}_{out}) can be seen as a set of points whose distance from any seed vertex is higher (lower) than a given threshold. More formally, denote by \mathbf{x} a generic point in \mathcal{H} and by \mathbf{x}_σ the position in \mathcal{H} of a generic seed vertex σ . Then, given two positive constants α and δ , s.t. $\delta \leq 1$ and $\alpha(1 + \delta) \leq 1$, we have:

$$\mathcal{H}_{\text{in}}(\alpha, \delta) = \left\{ \mathbf{x} \text{ s.t. } \max_{\sigma \in \mathcal{A}_0} \|\mathbf{x} - \mathbf{x}_\sigma\| \leq f^{-1}((1 + \delta)\alpha) \right\}$$

$$\mathcal{H}_{\text{out}}(\alpha, \delta) = \left\{ \mathbf{x} \text{ s.t. } \min_{\sigma \in \mathcal{A}_0} \|\mathbf{x} - \mathbf{x}_\sigma\| > f^{-1}((1 - \delta)\alpha) \right\}$$

where f is the non-increasing function defined in Section III. The two sub-regions are depicted in Fig. 4. Recall that, by construction, $|\mathcal{H}_{\text{in}}| = \Theta(C^k(n))$ since $f(d)$ vanishes for $d \gg C(n)$.

The theorem below proves that, given graph \mathcal{G}_1 (\mathcal{G}_2), it is possible to correctly distinguish nodes in $\mathcal{H}_{\text{in}}(\alpha, \delta)$ from nodes in $\mathcal{H}_{\text{out}}(\alpha, \delta)$ by counting the number of their neighbor seeds.

Theorem 2: Given a node $i \in \mathcal{G}_1$ ($i \in \mathcal{G}_2$), let S_i be the number of seeds that are neighbors of i on \mathcal{G}_1 (\mathcal{G}_2). We say that node i is accepted if $S_i > \alpha s K(n) a_0$. If $d_s = O(C(n))$ and $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$, then for an arbitrary $\delta > 0$,

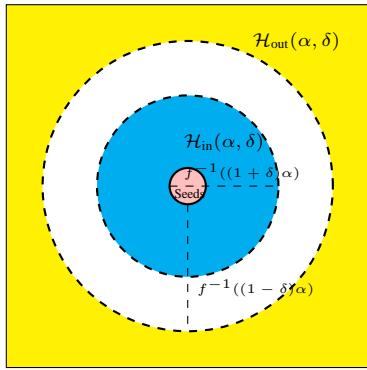


Fig. 4. Graphical representation of $\mathcal{H}_{\text{in}}(\alpha, \delta)$ and $\mathcal{H}_{\text{out}}(\alpha, \delta)$.

the above procedure correctly accepts all nodes located in $\mathcal{H}_{\text{in}}(\alpha, \delta)$ while it excludes all nodes located in $\mathcal{H}_{\text{out}}(\alpha, \delta)$.

Proof: See Appendix A. ■

Note that, in the above statement, $sK(n)$ is the probability that a node in \mathcal{G}_1 (\mathcal{G}_2)⁶ is connected with a seed node if their distance is $C(n)$ or less. Thus, $\alpha s K(n) a_0$ is a threshold on the number of connections between a node and the a_0 seed vertices.

Next, we denote by $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$, respectively, the set of nodes from \mathcal{G}_1 and \mathcal{G}_2 that are classified as located in $\mathcal{H}_{\text{in}}(\alpha, \delta)$. By construction, we have $|\mathcal{N}^1(\alpha)| = \Theta(nC^k(n))$ and $|\mathcal{N}^2(\alpha)| = \Theta(nC^k(n))$. We build the pairs graph $\mathcal{P}(\mathcal{N})$ that is induced by the nodes of \mathcal{G}_1 and \mathcal{G}_2 that belong to, respectively, $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$. While doing this, we make sure that a bad pair $[i_1, j_2]$ is included in $\mathcal{P}(\mathcal{N})$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also included in $\mathcal{P}(\mathcal{N})$. This is accomplished as follows. We apply the previous classification procedure twice, using two different values α_1 and α_2 , with $\alpha_1 > \alpha_2$, chosen in such a way that $\mathcal{H}_{\text{out}}(\alpha_1, \delta) \subseteq \mathcal{H}_{\text{in}}(\alpha_2, \delta)$. Then we insert in $\mathcal{P}(\mathcal{N})$ all pairs whose constituent nodes have been selected by at least one of the classification procedures, adding the constraint that at least one of the nodes must have been selected by both. Since by construction, no good pair $[i_1, i_2]$ exists s.t. i_1 falls in $\mathcal{H}_{\text{in}}(\alpha_1, \delta)$ and i_2 in $\mathcal{H}_{\text{out}}(\alpha_2, \delta)$ (or viceversa), the above condition is ensured.

We then apply the PGM algorithm on $\mathcal{P}(\mathcal{N})$. Our goal is now to verify that the conditions in Theorem 1 hold so that, applying the theorem and Corollary 1, we can claim that all good pairs in $\mathcal{P}(\mathcal{N})$ can be matched with no error. To this end, let us define $m = \Theta(nC^k(n))$, which in order sense equals the number of nodes in $\mathcal{N}^1(\alpha)$ and $\mathcal{N}^2(\alpha)$. Then recall that $p_{\min} = \Theta(p_{\max})$, $p_{\max} = K(n)$ and $K(n) = o(m^{-\gamma})$. Thus, for a sufficiently large r , $p_{\max} \ll m^{-\frac{3.5}{r}}$. Furthermore, since by assumption $nC^k(n)K(n) = \Omega(\log n)$, it follows $p_{\min} \gg m^{-1}$. At last, it is easy to see that $a_o/a_c \rightarrow \infty$. Indeed, from (1), $a_c = O(1/K(n))$ while, by assumption (see Theorem 2), $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$. In conclusion, we have that all good pairs whose nodes fall in $\mathcal{H}_{\text{in}}(\alpha_1, \delta)$ can be correctly matched.

To further expand the set of identified pairs, we can pursue the following simple approach. Starting from the bulk of pairs

⁶Recall that \mathcal{G}_1 (\mathcal{G}_2) is a subgraph obtained from \mathcal{G}_T by sampling the edges with probability s .

already matched, which act as seeds, we consider a larger region that includes the previous one. By properly setting a threshold r , we match all the pairs that have at least r neighbors among the seeds. So doing, we successfully match w.h.p. all good pairs in the region with no errors. More formally, the following theorem allows us to claim that our approach can be successfully employed.

Theorem 3: Consider a circular region centered in 0 and of radius ρ , $\mathcal{D}(0, \rho)$, with $\rho \geq C(n)$. Given that all (or almost all) nodes lying within $\mathcal{D}(0, \rho)$ have been correctly identified, it is possible to correctly identify (almost) all nodes in $\mathcal{D}(0, \rho_1) \setminus \mathcal{D}(0, \rho)$ with a probability $1 - o(n^{-1})$ for $\rho_1 = \rho + C(n)/2$ when $K(n) = o((nC^k(n))^{-\gamma})$ for some $\gamma > 0$. In addition, none of the bad pairs formed by nodes in $\mathcal{H} - \mathcal{D}(0, \rho)$ will be identified with a probability $1 - o(n^{-1})$. This is done by setting a threshold $r = \frac{n}{2}|\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}, C(n))| \frac{K(n)}{2}$, with $|\mathbf{x}| = \rho_1$ and identifying as good pairs those in $\mathcal{H} \setminus \mathcal{D}(0, \rho)$ that have at least r neighbors among good pairs in $\mathcal{D}(0, \rho)$. The proof is based on the application of standard concentration results, namely, Chernoff bound and inequalities in Appendix A. The detailed proof is given in Appendix A. Almost all good pairs can be matched w.h.p. by iterating the matching procedure of Theorem 3 a number of $\Theta(1/C(n))$ times. Indeed, each time the PGM algorithm successfully matches all good pairs whose constituent nodes lie within a distance $C(n)/2$ from the bulk of previously matched pairs. Note that Theorem 3 also guarantees that jointly over all rounds no bad pair is matched w.h.p.

VI. DENSE CLUSTERS

The case $K(n) = \omega((nC^k(n))^{-\gamma})$, for any $\gamma > 0$, is significantly different from the previous one since the de-anonymization algorithm must disregard all edges whose length is too short (shorter than a properly defined threshold $\omega(C(n))$) so as to avoid errors (i.e., matching bad pairs). The approach we propose to address this case relies on some results that we initially obtain for the special case in which \mathcal{G}_T is a bipartite graph. Then we extend such results to our clustered social graph and derive the minimum seed set size that is required for graph de-anonymization.

A. Results on bipartite graphs

Here we restrict our analysis to a ground-truth graph \mathcal{G}_T that is an $m_l \times m_r$ bipartite graph. Let \mathcal{M}_l denote the set of vertices on the left hand side (LHS), with $|\mathcal{M}_l| = m_l$, and \mathcal{M}_r the set of vertices on the right hand side (RHS), with $|\mathcal{M}_r| = m_r$. We assume that for any pair of vertices $i \in \mathcal{M}_l$ and $j \in \mathcal{M}_r$ an edge (i, j) exists in the graph with probability p_{ij} , with $p_{\min} \leq p_{ij} \leq p_{\max}$ and $p_{\max} = \eta p_{\min}$ for some finite positive η . The goal here is to identify a minimum number of seeds a_0 , with $a_0 = |\mathcal{A}_0^l|$ in \mathcal{M}_l and $a_0 = |\mathcal{A}_0^r|$ in \mathcal{M}_r , such that vertices in \mathcal{M}_l and \mathcal{M}_r can be correctly matched.

Let us first consider the case where $m_l = m_r = m$, for which the theorem below holds.

Theorem 4: Assume that \mathcal{G}_T is an $m \times m$ bipartite graph and that two sets of seeds, \mathcal{A}_0^l and \mathcal{A}_0^r , of cardinality a_0 , are available on, respectively, the LHS and the RHS of the

graph. Then the PGM algorithm with threshold $r \geq 4$ correctly identifies $m - o(m)$ good pairs w.h.p. on the RHS and the LHS of graph $\mathcal{P}(\mathcal{G}_T)$, with no errors, if:

1. $m^{-1} \ll p_{\min} \leq p_{\max} \leq m^{-\frac{3.5}{r}}$
2. $\liminf_{m \rightarrow \infty} a_0/a_c > 1$

$$\text{where } a_c = \left(1 - \frac{1}{r}\right) \left(\frac{(r-1)!}{m(p_{\min}s^2)^r} \right)^{\frac{1}{r-1}}.$$

Proof: See Appendix A. ■

Theorem 4 can be extended to the more general case where $m_l \neq m_r$, as shown by the corollary below.

Corollary 2: Assume that \mathcal{G}_T is an $m_l \times m_r$ bipartite graph and define $m = \min(m_l, m_r)$. Under the same assumptions of Theorem 4, the PGM algorithm with threshold $r \geq 4$ successfully identifies w.h.p. $m - o(m)$ good pairs on both the LHS and the RHS of $\mathcal{P}(\mathcal{G}_T)$, with no errors. Furthermore, the PGM algorithm can be successfully applied to an imperfect pairs graph $\hat{\mathcal{P}}(\mathcal{G}_T) \subset \mathcal{P}(\mathcal{G}_T)$ comprising a finite fraction of pairs on both the LHS and the RHS of $\mathcal{P}(\mathcal{G}_T)$ and satisfying the following constraint: a bad pair $[i_1, j_2] \in \mathcal{P}(\mathcal{G}_T)$ is included in $\hat{\mathcal{P}}(\mathcal{G}_T)$ only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also in $\hat{\mathcal{P}}(\mathcal{G}_T)$.

Proof: The assertion can be proved by following the same arguments as in Theorem 4 and applying Corollary 1. ■

Finally, we prove the following result, which shows that all good pairs can be matched with no errors w.h.p.

Theorem 5: Consider that \mathcal{G}_T is an $m_l \times m_r$ bipartite graph with $m_l = \omega(\sqrt{m_r})$ and that a seed set \mathcal{A}_0^l is available on the LHS of the graph, with $|\mathcal{A}_0^l| = a_0 = \Theta(m_l)$. With probability larger than $1 - e^{-\frac{m_l}{\sqrt{m_r}}}$, all the m_r good pairs on the RHS can be successfully identified with no errors, provided that:

1. $\frac{1}{\sqrt{m_r}} \ll p_{\min} \leq p_{\max} \ll 1$
2. $p_{\min} = \Theta(p_{\max})$
3. a matching algorithm is used on $\mathcal{P}(\mathcal{G}_T)$ that matches all pairs on the RHS that have at least r adjacent seeds on the LHS, with $r = a_0 \frac{p_{\min}}{2}$.

The same result holds in case of imperfect pairs graph comprising a finite fraction of all possible pairs on the RHS.

Proof: Without loss of generality, we assume $a_0 \geq cm_r$ for some $c > 0$. The proof is obtained by applying the inequalities reported in Appendix A. First, observe that, given a good pair $[j_1, j_2]$ on the RHS of the pairs graph, its number of adjacent seeds on the LHS is $E[N_g] \geq a_0 p_{\min} = 2r$. Thus, by applying inequality (7) and union bound, we have:

$$P(\text{all good pairs on the RHS have at least } r \text{ adjacent seeds}) \geq 1 - m_r e^{-cm_l p_{\min} H(\frac{1}{2})} \geq 1 - e^{-\frac{m_l}{\sqrt{m_r}}}$$

which imply that all good pairs on the RHS are successfully matched since $m_l = \omega(\sqrt{m_r})$. Similarly, considering a bad pair $[j_1, k_2]$ on the RHS, its number of adjacent seeds on the LHS is $E[N_b] \leq cm_r(p_{\max})^2 \ll r$. Thus, by applying inequality (9) and union bound, we have:

$$P(\text{all bad pairs on the RHS have less than } r \text{ adjacent seeds}) \geq 1 - m_r^2 e^{-cm_l \frac{p_{\min}}{4} \log\left(\frac{p_{\min}}{(p_{\max})^2}\right)} \geq 1 - e^{-\frac{m_l}{\sqrt{m_r}}}.$$

■

B. The de-anonymization procedure

We now outline how our proposed de-anonymization technique works. First, we consider two hyper-cubic regions, $\mathcal{H}_l \subset \mathcal{H}$ and $\mathcal{H}_r \subset \mathcal{H}$, whose side is $h(n) = \Omega(C(n))$ and whose distance is $g(n) = \omega(C(n))$ (see Fig. 3). Note that by construction, given two vertices $i \in \mathcal{H}_l$ and $j \in \mathcal{H}_r$, $p_{\min} = K(n)f(g(n) + \sqrt{kh(n)}) \leq p_{ij} \leq K(n)f(g(n)) = p_{\max}$. Let us assume $p_{\max} = \eta p_{\min}$ for some constant $\eta > 1$.

We then extract vertices in \mathcal{H}_l and \mathcal{H}_r from the rest of vertices so that we can focus on the bipartite graph induced by the nodes in the two sub-regions, along with the edges between them. To this end, we assume that two sufficiently large sets of seeds are available in \mathcal{H}_l and \mathcal{H}_r so that Theorem 2 can be applied. In this regard, observe that we can use the same procedure as in Section V, to make sure that a bad pair $[i_1, j_2]$ is included in the pair graph only if either $[i_1, i_2]$ or $[j_1, j_2]$ are also included in it. We can then apply Corollary 2.

It follows that the execution of the PGM algorithm ensures that almost all of the good pairs in either the LHS or the RHS of the pairs graph are correctly de-anonymized. Without lack of generality, we assume that almost all pairs on LHS are de-anonymized, i.e., $m_l < m_r$, and that a non-negligible fraction of the good pairs on the RHS have still to be identified. Then the rest of good pairs on the RHS can be matched by applying Theorem 5.

To further extend the de-anonymization procedure, we first observe that it is possible to estimate in order sense the length of the edges between two nodes, again, by exploiting the dense structure of the clusters.

Proposition 1: Given two nodes in region \mathcal{H} , it is possible to estimate with arbitrary precision their mutual distance d as far as $d \ll C(n)(nK^2(n)C^k(n))^{\frac{1}{\beta}}$.

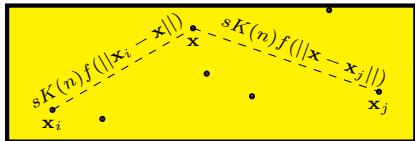


Fig. 5. Computation of $\mathbb{E}[N_{ij}]$.

Proof: Let us consider two nodes i and j on \mathcal{G}_1 (\mathcal{G}_2) whose mutual distance is d_{ij} . Let N_{ij} be the variable that represents the number of their common neighbors. By construction, we have:

$$\begin{aligned}\mathbb{E}[N_{ij}] &= (n-2)s^2K^2(n) \int_{\mathcal{H}} f(||\mathbf{x} - \mathbf{x}_i||)f(||\mathbf{x} - \mathbf{x}_j||)d\mathbf{x} \\ &= \Theta(nC^k(n)K^2(n)f(d_{ij})).\end{aligned}$$

Observe that $\mathbb{E}[N_{ij}]$ is continuous and strictly decreasing with d_{ij} , and thus invertible. Now, applying Chernoff bound we can show that for any $0 < \delta < 1$

$$\mathbb{P}\left(\frac{|N_{ij} - \mathbb{E}[N_{ij}]|}{\mathbb{E}[N_{ij}]} > \delta\right) \leq e^{-c(\delta)\mathbb{E}[N_{ij}]}$$

for a proper constant $c(\delta) > 0$. Furthermore for $\delta > 1$

$$\mathbb{P}\left(\frac{N_{ij}}{\mathbb{E}[N_{ij}]} > \delta\right) \leq e^{-c(\delta)(\delta\mathbb{E}[N_{ij}]\log\delta)}$$

Since $\mathbb{E}[N_{ij}] \rightarrow \infty$ as long as $d \ll C(n)(nK^2(n)C^k(n))^{\frac{1}{\beta}}$, the assertion follows. \blacksquare

We can therefore use the number of common neighbors between two endpoint nodes as an estimator for their distance. We then set two thresholds, $d_L = \Theta(C(n)\log(n^{1/k}C(n)))$ and $d_H = \lambda d_L$ (with $\lambda > 1$), and we leverage the above result to correctly classify the edges departing from previously matched nodes into three categories: edges that are shorter than d_L , edges that are longer than d_H and edges of length comprised between d_L and d_H . In particular, we are interested in the latter, for which the following result holds.

Proposition 2: Assume $K(n) = \omega((nC^k(n))^{-\gamma}) \forall \gamma > 0$. Consider a set comprising a finite fraction of the nodes of \mathcal{G}_1 (\mathcal{G}_2) that lie in a region of side $\Theta(C(n))$, and the edges incident to them. For an arbitrarily selected $\delta > 0$, w.h.p (i.e., with a probability larger than $1 - [C(n)]^k$) we can select all edges whose length d is $(1 + \delta)d_L \leq d \leq (1 - \delta)d_H$. Furthermore, no edges whose length $d < (1 - \delta)d_L$ and $d > (1 + \delta)d_H$ are selected.

Proof: The proof follows the same scheme of proof of Theorem 2, here we provide just a sketch.

Fix a $\delta > 0$, first we consider all edges whose length does not exceed $(1 - \delta)d_L$. By applying Proposition 1 and the union bound, the probability that they are jointly not selected can be bounded by:

$$\begin{aligned}P(\text{some edge with length } d < (1 - \delta)d_L \text{ is selected}) \\ \leq N_e e^{-c'nC^k(n)K^2(n)f((1-\delta)d_L)}\end{aligned}$$

where N_e is the number of edges with length $d < (1 - \delta)d_L$ and c' is an opportune constant. Now since by construction $N_e = O(nC^k(n)D(n)) = O((nC^k(n))^2K(n))$ and $d_L = \Theta(C(n)\log C(n))$ none of those edges is included. Similarly we can show that all edges whose length is $(1 + \delta)s_L \leq d \leq (1 - \delta)$ are selected.

To show that none of the edges whose length is exceeding $d_H(1 + \delta)$ are selected we resort on the same ideas of the proof of Theorem 2. In particular, we partition such edges into smaller groups containing only those edges of similar length. For each of groups we have defined, we exploit Chernoff inequality along with the union bound (similarly as before) to provide an upper bound to the probability that at least one of such edges is selected. We can conclude our proof showing that previous property holds uniformly on all the groups. \blacksquare

At this point, we consider a bipartite graph whose LHS is still represented by \mathcal{H}_l , and whose RHS is given by the nodes that are connected with those in \mathcal{H}_l through edges of length comprised between d_L and d_H . We can therefore apply Theorem 5 and match w.h.p. all good pairs on the RHS, with no errors. The procedure is then iterated so as to successfully de-anonymize the whole network graph. Note that, at every step we apply the following proposition to extract a group of matched nodes whose mutual distance is $\Theta(C(n))$.

Proposition 3: Assume $K(n) = \omega((nC^k(n))^{-\gamma}) \forall \gamma > 0$. Given a node i , we can set a threshold $d_T = \Theta(C(n))$ and select all nodes in \mathcal{G}_1 (\mathcal{G}_2) whose estimated distance from i is less than d_T . So doing, for an arbitrarily selected $\delta > 0$, we successfully select with a probability larger than $1 - [C(n)]^k$

all nodes whose real distance is $d \leq (1 - \delta)d_T$. Furthermore, no nodes whose distance from i is $d > (1 + \delta)d_T$ are selected by our algorithm.

Proof: The proof follows exactly the same lines as the proof of Proposition 2. ■

C. Minimum seed set size

To explicitly derive the minimum size of the seed set, we need to further specify $h(n)$ and $g(n)$, which are to be carefully selected so as to minimize the resulting critical size a_c in Theorem 4 and Corollary 2.

Starting from the result provided by Theorem 4, a_c can be written as:

$$\begin{aligned} a_c &= \left(1 - \frac{1}{r}\right) \left(\frac{(r-1)!}{m(p_{\min}s^2)^r}\right)^{\frac{1}{r-1}} \\ &\leq \left(\frac{r-1}{(m(p_{\min}s^2)^{\frac{1}{r-1}} p_{\min}s^2)}\right) \leq \frac{r-1}{p_{\min}s^2}. \end{aligned} \quad (6)$$

The above expression can be minimized by maximizing p_{\min} , i.e., by minimizing $g(n)$ (recall that $p_{\min} = K(n)f(g(n)) + \sqrt{kh(n)}$). However, $g(n)$ and $h(n)$ must also be selected in such a way that condition 1) of Theorem 4 is met. Additionally, as mentioned, it must be ensured that $h(n) = \Omega(C(n))$. At last, by standard concentration results, m_l and m_r turn out to be both $\Theta(nh^k(n))$ provided that $h(n) \geq (\log n/n)^{1/k}$.

Previous considerations induce us to fix $h(n) = \Theta(C(n)) \geq (\log n/n)^{1/k}$ (i.e., the minimum possible value in order sense), which corresponds to have $m = \Theta(nC^k(n))$ (recall that $m = \min(m_l, m_r)$). We then derive $g(n)$ by forcing $p_{\max} \approx m^{-\frac{\alpha}{r}}$, with $3.5 < \alpha < 4$ and $r \geq 4$. Note that condition 1) of Theorem 4 is met since p_{\max} and p_{\min} are both $\Theta(m^{-\frac{\alpha}{r}})$. Hence, we have $p_{\max} = \Theta((nC^k(n))^{-\frac{\alpha}{r}})$ and $g(n) = \Theta(n^{\frac{\alpha}{\beta r}} [C(n)]^{1+\frac{\alpha k}{\beta r}} [K(n)]^{\frac{1}{\beta}})$.

Given the above expression for p_{\max} , considering that $p_{\max} = \eta p_{\min}$ and using (6), the minimum seed set size can be made as small as

$$a_c = O([nC^k(n)]^\epsilon)$$

for any $\epsilon > 0$, by choosing $r > \frac{4}{\epsilon}$.

Finally, we remark that the obtained a_c is in order sense greater than the minimum number of seeds needed to apply Theorem 2 while selecting nodes in regions \mathcal{H}_l and \mathcal{H}_r , thus the whole construction is consistent.

VII. EXPERIMENTAL VALIDATION

Although our results hold asymptotically as $n \rightarrow \infty$, we can expect to qualitatively observe the main effects predicted by the analysis also in finite-size graphs. We will first investigate the performance of graph matching algorithms in synthetic graphs generated according to our model of clustered networks, and then apply them to real social network graphs.

A. Synthetic graphs

In this section we consider bi-dimensional graphs having $n = 10,000$, the sampling probability $s = 0.8$ and, unless

otherwise specified, the average node degree in the ground-truth graph $D(n) = 30$.

Fig. 6 reports the average number of correctly matched nodes across 1,000 runs of the PGM algorithm (using $r = 5$) in various cases, as function of the number of seeds. In each run, seeds are either chosen uniformly at random among all nodes (label ‘uniform seeds’), or as a compact set around one randomly chosen seed (label ‘compact seeds’). In our model of clustered graphs, we have fixed $\beta = 3$ (the decay exponent of the edge probability beyond $C(n)$), and we consider either $K(n) = 0.05$ or $K(n) = 0.2$. As reference, in the plot we also show the phase transition occurring (at about 600 seeds) when \mathcal{G}_T is a $G(n, p)$ graph having the same average node degree. The plot confirms the wave-like nature of the identification process as predicted by our analysis, namely: i) clustered networks (larger $K(n)$) can be matched starting from a much smaller seed set as compared to $G(n, p)$; ii) such huge reduction requires seeds to be selected within a small sub-region of \mathcal{H} .

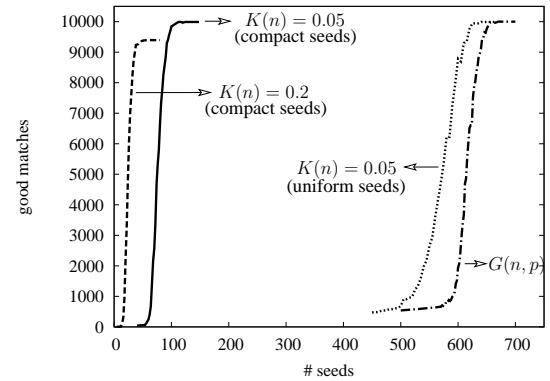


Fig. 6. Comparison of PGM performance (with $r = 5$) in different networks with $n = 10,000$. Number of good matches (averaged over 1,000 runs) as a function of the number of seeds, chosen either uniform or compact.

What the plot in Fig. 6 does not clearly show (except for a rough estimate based on the maximum number of correctly matched nodes) is the error ratio incurred by the PGM algorithm, which is expected to become larger and larger as we increase the level of clustering in the network. This phenomenon is confirmed by Fig. 7, which reports the average error ratio (bad matches over all matches) incurred by PGM as a function of $K(n)$, starting from a compact set of seeds. In Fig. 7 we have considered also different values of β . The little circle denotes the operating point already considered for the left-most curve in Fig. 6, having an error ratio of about 5%. The plot reveals that the error ratio increases dramatically when $K(n)$ tends to 1, confirming that PGM cannot be safely applied in highly clustered networks. The effect of β is more intriguing: smaller β 's produce fewer errors since generated network graphs tend to become more similar to $G(n, p)$, where PGM is known to perform very well. As side-effect, smaller values of β tend to slightly increase the percolation threshold (not shown in the plot). For example, for $K(n) = 0.4$, the critical number of seeds (estimated from simulations) corresponding to $\beta = 2.2, 2.5, 3, 4$ are equal to 11, 15, 24, 45, respectively.

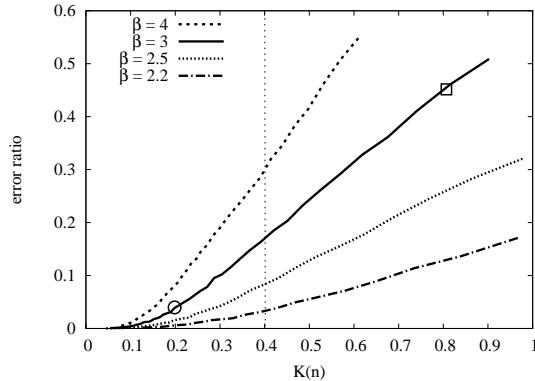


Fig. 7. Error ratio of PGM as a function of $K(n)$ for different values of β , starting from compact seeds.

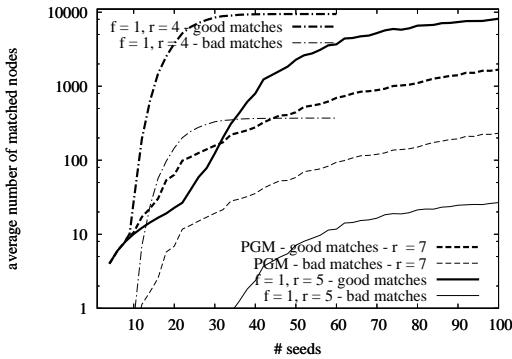


Fig. 8. Average number of good and bad pairs matched by different algorithms for $K(n) = 0.8$, $\beta = 3$, starting from compact seeds.

Next, we focus on the ‘hard’ case corresponding to the little square shown in Fig. 7, i.e., $K(n) = 0.8$, $\beta = 3$. This case corresponds to networks having highly dense clusters, where the performance of the original PGM algorithm is rather poor (error ratio about 50%). Fig. 8 shows the average number of nodes matched by different algorithms as a function of the number of seeds: thick lines correspond to good matches, whereas thin lines (with the same line style) refer to bad matches produced by a given algorithm. For sake of simplicity, network de-anonymization is performed by applying a simplified version of the algorithm proposed and analysed in Section VI. This simple algorithm consists in adopting PGM after having removed all graph edges shorter than $x \cdot C(n)$. In the following, we will call this algorithm ‘filtered PGM’ and we will label the corresponding curves in the plots by ‘ $f = <x>$ ’. We stress that filtered PGM provides qualitatively similar results to the performance of the algorithm in Section VI.

Looking at Fig. 8, it is important to remark that in this scenario the performance of the various algorithms is highly sensitive to the location of the set of seeds (in each run we uniformly select one seed among all nodes, and put all of the other seeds around it). Since we average the results over 1,000 runs, this explains why all curves do not exhibit a sharp

TABLE II
COMBINATIONS OF PARAMETERS ACHIEVING ERROR RATIO 3%,
PERCOLATION PROBABILITY 50%

average node degree	f	# seeds
36	1.1	22
45	1.2	24
53	1.3	28
64	1.4	32

transition⁷. An average number of matched nodes equal to, say, 2,000, must be given the following probabilistic interpretation: about 1/5 of (uniformly chosen) initial locations allow us to match almost all nodes (10,000), while 4/5 of initial locations do not trigger the percolation effect.

Also, we note that the poor performance of standard PGM cannot be fixed by just increasing the threshold r : using $r = 7$, PGM still produces about 12% error ratio, while requiring many more seeds (only about 2,000 nodes are matched on average starting from 100 seeds). Instead, filtered PGM, with $f = 1$ and $r = 4$, requires very few seeds to match almost all nodes, incurring about 3.7% error ratio. Using $f = 1$, $r = 5$, filtered PGM requires more seeds, but achieves as low as 0.3% error ratio.

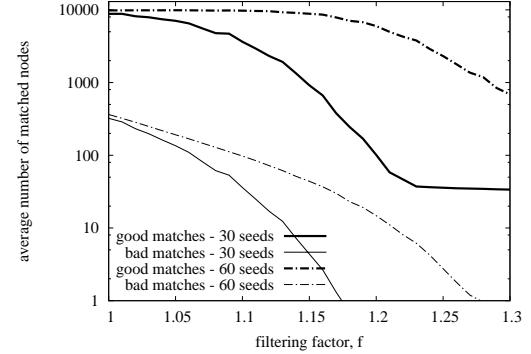


Fig. 9. Effect of varying the filtering factor f for fixed $r = 4$ (scenario with $K(n) = 0.8$).

Next, we fix r and increase the filtering factor f so as to diminish the number of errors while, however, reducing the average number of matched nodes (i.e., the probability to trigger percolation from a given seed set). Fig. 9 illustrates this effect for $r = 4$, in the case of two different seed set sizes, 30 and 60. Having 60 seeds one could, for example, employ $f = 1.1$ obtaining very high chance of percolation (almost 100%) and small error ratio (around 1%).

Alternately, we can fix a desired error ratio and average number of matched nodes (i.e., the probability to trigger large-scale percolation), and look for the filtering factor and number of seeds that let us achieve these goals. Table VII-A reports an example of this numerical exploration, in which we vary the average degree of the nodes in \mathcal{G}_T corresponding to each examined scenario (the average degree can be increased, for fixed $K(n) = 0.8$, by increasing $C(n)$). The results in

⁷We verified that, if we instead fix the very first seed across all runs, a sharp transition appears. However, the transition threshold changes as we vary the initial seed (results not shown here).

Table VII-A validate, at least qualitatively, the counter-intuitive theoretical predictions in Table IV: as we increase $C(n)$ (and thus the average node degree), the seed set size necessary to achieve a desired matching performance increases as well.

B. Real social graphs

We consider a real graph derived from the Slovak social network Pokec [12]. The public data set, available at [13], is a directed graph with 1,632,803 vertices, where nodes are users of Pokec and directed edges represent friendships. Since the original graph contains too many vertices for our computational power, and since we would like to isolate the impact of clustering from the effect of long-tailed degree distributions, we considered only vertices having: i) in-degree larger than 20; ii) out-degree smaller than 200. We ended up with a reduced graph having $n = 133,573$ nodes, average (in or out) degree 40.8 and clustering coefficient 0.1. We use this graph as our ground-truth, and employ an edge sampling probability $s = 0.8$. Notice that we maintain the direct nature of the edges, since all considered algorithms immediately apply to direct networks as well⁸.

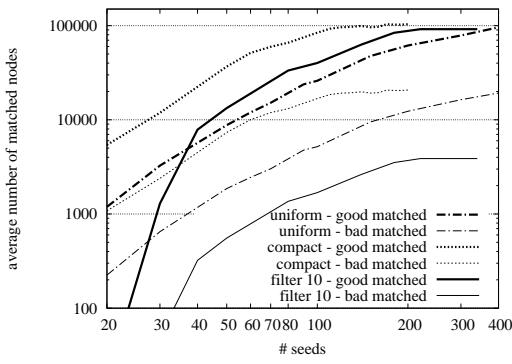


Fig. 10. Performance of matching algorithms in a subset of the friendship graph of the social network Pokec.

Fig. 10 shows the performance of the different algorithms using threshold $r = 6$. As before, curves labelled ‘uniform’ refer to the PGM algorithm in which seeds are selected uniformly at random among the nodes. Curves labelled ‘compact’ refer to the PGM algorithm in which seeds are chosen among the closest neighbors of a uniformly selected node. Curves labelled ‘filter 10’ differ from the previous one in that the edges connecting each node to its nearest 10 neighbors are not used by the algorithm. We emphasize that a $G(n, p)$ having the same number of nodes and average degree would require $a_c = 5,783$ seeds, according to (1). In contrast, all considered algorithms require much fewer seeds to match almost all nodes, confirming that real social networks are much simpler to de-anonymize than $G(n, p)$. In particular, the uniform variant requires about 300 seeds to match on average more than 100,000 nodes, but incurs a quite large error ratio (about 17%). The compact variant reduces this number roughly by a factor 3, but produces the same error ratio. At

⁸In direct networks, counters of matchable pairs are incremented only by using outgoing edges from matched pairs.

last, the filtered variant requires slightly more seeds than the compact one, but it allows to lower down the error ratio to about 4%. The above results confirm the crucial performance improvement that can be obtained by jointly: i) starting from a compact set of seeds (to exploit the wave-propagation effect), ii) carefully discarding edges connecting nodes to their local clusters (to limit the errors).

VIII. CONCLUSIONS

We focused on the effect of node clustering on social graph de-anonymization. We defined a general model for network graphs that can represent different levels of node clustering. Then we designed de-anonymization algorithms and analysed their performance by using bootstrap percolation. Our theoretical results highlight that clustering significantly helps to reduce the minimum seed set size required for network de-anonymization, and that our algorithms can successfully limit the error rate of the de-anonymization procedure. Our findings were confirmed by numerical experiments on synthetic and real social graphs.

APPENDIX

Lemma 1: Let $H(b) = 1 - b - b \log b$ for $b > 0$. Suppose $n \in \mathbb{N}$ $p \in (0, 1)$ and $0 \leq k \leq n$ let $\mu = np$ if $k \leq \mu$ then:

$$P(\text{Bin}(n,p) \leq k) \leq \exp\left(-\mu H\left(\frac{k}{\mu}\right)\right) \quad (7)$$

if $k > \mu$ then:

$$P(\text{Bin}(n,p) \geq k) \leq \exp\left(-\mu H\left(\frac{k}{\mu}\right)\right) \quad (8)$$

if $k > e^2\mu$ then

$$P(\text{Bin}(n,p) \geq k) \leq \exp\left(-\frac{k}{2} \log \frac{k}{\mu}\right) \quad (9)$$

Without loss of generality, let us focus on \mathcal{G}_1 and let us consider a node $i \in \mathcal{H}_{\text{in}}(\alpha, \delta)$. By construction, the number of seeds that are neighbors of i on \mathcal{G}_1 is given by $S_i = \sum_{\sigma \in \mathcal{A}_0} X_{i\sigma} S_{i\sigma}^1 \geq_{st} Y_i \geq_{st} Y$ where

$$Y_i = \text{Bin}(a_0, sK(n)f(\max_{\sigma \in \mathcal{A}_0} \|\mathbf{x}_i - \mathbf{x}_\sigma\|))$$

and $Y = \text{Bin}(a_0, sK(n)(1 + \delta)\alpha)$, with $\mathbb{E}[Y] = sK(n)(1 + \delta)\alpha a_0$. Now, using the inequalities reported in Appendix A, we can bound:

$$\begin{aligned} P(Y_i < \alpha sK(n)a_0) &\leq \exp\left(-\mathbb{E}[Y_i]H\left(\frac{\alpha sK(n)a_0}{\mathbb{E}[Y_i]}\right)\right) \\ &\leq \exp\left(-(1 + \delta)\alpha sK(n)a_0 H\left(\frac{1}{1 + \delta}\right)\right) \end{aligned} \quad (10)$$

with $H(b) = 1 - b + b \log b$.

If we consider jointly all nodes in $\mathcal{H}_{\text{in}}(\alpha, \delta)$ and we denote with N_{in} their number, we can bound the probability that every node in $\mathcal{H}_{\text{in}}(\alpha, \delta)$ is accepted with:

$$\begin{aligned} P(\text{all nodes in } \mathcal{H}_{\text{in}} \text{ are accepted} \mid N_{\text{in}}) \\ \leq 1 - N_{\text{in}} \exp\left(-(1 + \delta)\alpha sK(n)a_0 H\left(\frac{1}{1 + \delta}\right)\right), \end{aligned} \quad (11)$$

with (11) that tends to 1 if $\log N_{\text{in}} = (1 + \delta)\alpha sH\left(\frac{1}{1+\delta}\right)K(n)a_0 \rightarrow -\infty$. This can be enforced by opportunely setting $a_0 = \Omega\left(\frac{\log N_{\text{in}}}{K(n)}\right)$. Since by construction $|\mathcal{H}_{\text{in}}| > C^k(n) \geq \frac{\log n}{n}$, we have w.h.p. $N_{\text{in}} \leq 2n|\mathcal{H}_{\text{in}}|$ by standard concentration results (See Lemma 2). As a consequence, w.h.p.

$$P(\text{all vertices in } \mathcal{H}_{\text{in}} \text{ are accepted}) \rightarrow 1$$

provided that a_0 is opportunely chosen, with:
 $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$.

Then we focus on the nodes in $\mathcal{H}_{\text{out}}(\alpha, \delta)$ and we show that all those nodes are jointly rejected. Conceptually we repeat the same approach as before, however, the argument is made slightly more complex by the fact that, to achieve tight bounds on the probability that all nodes in $\mathcal{H}_{\text{out}}(\alpha, \delta)$ are jointly rejected, we need to partition $\mathcal{H}_{\text{out}}(\alpha, \delta)$ into smaller sub-regions containing nodes, which lie at similar distance from the seeds.

Assuming $\delta < \frac{e^2-1}{e^2}$, we define $\mathcal{H}_{\text{out}}^1 = \mathcal{H}^1(\alpha, \frac{e^2-1}{e^2}) \subset \mathcal{H}_{\text{out}}(\alpha, \delta)$ and $\mathcal{H}_{\text{out}}^0 = \mathcal{H}_{\text{out}}(\alpha, \delta) \setminus \mathcal{H}_{\text{out}}^1$. Furthermore, we partition $\mathcal{H}_{\text{out}}^1$ into disjoint sub-regions, i.e., $\mathcal{H}_{\text{out}}^1 = \cup_{h \geq 1} \mathcal{H}_{\text{out}}^{1,h}$, with $\mathcal{H}_{\text{out}}^{1,h} = \mathcal{H}_{\text{out}}(\frac{\alpha, h^\beta e^2 - 1}{h^\beta e^2}) \setminus \mathcal{H}_{\text{out}}(\alpha, \frac{(h+1)^\beta e^2 - 1}{(h+1)^\beta e^2})$. Now, given a vertex i in $\mathcal{H}_{\text{out}}^0$ ($\mathcal{H}_{\text{out}}^{1,h}$), the number of its neighbor seeds S_i on \mathcal{G}_1 can be bounded from above by a $\text{Bin}(a_0, sK(n)(1-\delta)\alpha) \left(\text{Bin}(a_0, \frac{sK(n)}{h^\beta e^2}\alpha) \right)$. Furthermore, by elementary geometrical arguments, it can be shown that: i) $|\mathcal{H}_{\text{out}}^0| = \Theta(C^k(n))$, ii) $|\mathcal{H}_{\text{out}}^{1,1}| = \Theta(C^k(n))$ and iii) $\mathcal{H}_{\text{out}}^{1,h} = \Theta(h^{k-1}\mathcal{H}_{\text{out}}^{1,1})$.

Denoted with N_{out}^0 and $N_{\text{out}}^{1,h}$ the number of nodes in $\mathcal{H}_{\text{out}}^0$ and $\mathcal{H}_{\text{out}}^{1,h}$, respectively, by exploiting again the inequalities in Appendix A w.h.p. we have:

$$\begin{aligned} P(\text{all nodes in } \mathcal{H}_{\text{out}}^0 \text{ are rejected}) &\leq \\ 1 - N_{\text{out}}^0 \exp\left(-(1-\delta)\alpha sK(n)a_0 H\left(1-\delta\right)\right) &\rightarrow 1. \end{aligned}$$

The above expression holds under the assumption that $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$. Indeed, we remark that $N_{\text{out}}^0 \leq 2n|\mathcal{H}_{\text{out}}^0| = \Theta(nC^k(n))$ w.h.p. At last,

$$\begin{aligned} P(\text{all nodes in } \mathcal{H}_{\text{out}}^1 \text{ are rejected}) &\leq \\ 1 - \sum_{h=1}^{\infty} N_{\text{out}}^{1,h} \exp\left(-\frac{\alpha sK(n)a_0}{2}(\beta \log h + 2)\right) &. \end{aligned}$$

For every h , $N_{\text{out}}^{1,h} \leq 2n|\mathcal{H}^{1,h}| = \Theta(nh^{k-1}C^k(n))$; also, the number of sub-regions of $\mathcal{H}_{\text{out}}^1$ is $O(n/C^k(n))$. Thus, w.h.p we have that jointly on all h 's, the number of nodes in these sub-regions can be bounded by $2n|\mathcal{H}^{1,h}|$. Under the assumption that $a_0 = \Omega\left(\frac{\log(nC^k(n))}{K(n)}\right)$, it can be easily shown that $P(\text{all nodes in } \mathcal{H}_{\text{out}}^1 \text{ are rejected}) \rightarrow 1$.

The following proof uses some notation that has been introduced in [6] and that here is omitted for brevity (the reader may also refer to Appendix A for a more detailed description of the PGM algorithm and associated notation).

For any two vertices $i \in \mathcal{M}_l$ and $j \in \mathcal{M}_r$, let X_{ij} be the Bernoulli random variable that represents the presence of

an edge $(i, j) \in \mathcal{E}$. By construction, $Ber(p_{\min}) \leq_{st} X_{ij} \leq_{st} Ber(p_{\max})$. I.e., two variables \underline{X}_{ij} and \overline{X}_{ij} , with distribution, respectively, $Ber(p_{\min})$ and $Ber(p_{\max})$, can be defined on the same probability space as X_{ij} such that $\underline{X}_{ij} \leq X_{ij} \leq \overline{X}_{ij}$ point-wise.

We consider the corresponding pairs graph $\mathcal{P}(\mathcal{G}_T)$, which is, by construction, composed of all the pairs of vertices residing in \mathcal{M}_l and \mathcal{M}_r and of the edges connecting pairs of vertices in \mathcal{M}_l with pairs of vertices in \mathcal{M}_r . We denote by \mathcal{P}_l and \mathcal{P}_r , respectively, the set of pairs of $\mathcal{P}(\mathcal{G}_T)$, whose vertices lie in \mathcal{M}_l and \mathcal{M}_r . Observe that, given two good pairs $[i_1, i_2] \in \mathcal{P}_l$ and $[j_1, j_2] \in \mathcal{P}_r$, the presence of an edge in $\mathcal{P}(\mathcal{G}_T)$ is associated with the random variable:

$$Y_{[i_1, i_2], [j_1, j_2]} = X_{ij} X_{ij} S_{ij}^1 S_{ij}^2 = X_{ij}^2 S_{ij}^1 S_{ij}^2$$

where S_{ij}^1 and S_{ij}^2 are mutually independent $Ber(s)$ r.v's, which are in turn independent of X_{ij} . By construction, $p_{\min}s^2 \leq \mathbb{E}[Y_{[i_1, i_2], [j_1, j_2]}] \leq p_{\max}s^2$. Instead, given two bad pairs $[i_1, k_2] \in \mathcal{P}_l$ and $[j_1, l_2] \in \mathcal{P}_r$, $Y_{[i_1, k_2], [j_1, l_2]} = X_{ik} X_{kl} S_{ik}^1 S_{kl}^2$, with $p_{\min,s}^2 \leq \mathbb{E}[Y_{[i_1, k_2], [j_1, l_2]}] \leq p_{\max,s}^2$. Finally, if we consider one good pair and one bad pair (e.g., $[i_1, i_2] \in \mathcal{P}_l$ and $[j_1, k_2] \in \mathcal{P}_r$), $Y_{[i_1, i_2], [j_1, k_2]} = X_{ik} X_{ik} S_{ik}^1 S_{ik}^2$, with $p_{\min,s}^2 \leq \mathbb{E}[Y_{[i_1, i_2], [j_1, k_2]}] \leq p_{\max,s}^2$.

Recall that we assume that two seed sets, $\mathcal{A}_0^l \in \mathcal{P}_l$ and $\mathcal{A}_0^r \in \mathcal{P}_r$ (with $|\mathcal{A}_0^l| = |\mathcal{A}_0^r|$), are available. On $\mathcal{P}(\mathcal{G}_T)$ we run the PGM algorithm [6], opportunely modified, as follows. At every time step t , we extract uniformly at random one pair $\mathbf{z}^l(t) = [z_1^l, z_2^l]_t \in \mathcal{A}_{t-1}^l \setminus \mathcal{Z}_{t-1}^l$ and $\mathbf{z}^r(t) = [z_1^r, z_2^r]_t \in \mathcal{A}_{t-1}^r \setminus \mathcal{Z}_{t-1}^r$, adding a mark to all the neighbor pairs in \mathcal{P}_r and \mathcal{P}_l , respectively. In other words, matched pairs in \mathcal{P}_l contribute to the mark of pairs in \mathcal{P}_r and vice versa. Thus, for a generic node pair $[i_1, j_2] \in \mathcal{P}_r \setminus \mathcal{Z}_t^r$, marks are updated according to the iteration: $M_{[i_1, j_2]}^r(t) = M_{[i_1, j_1]}^r(t-1) + Y_{\mathbf{z}^l(t), [i_1, j_2]}$. Similarly, for $[i_1, j_2] \in \mathcal{P}_l$ marks are updated according to $M_{[i_1, j_2]}^l(t) = M_{[i_1, j_2]}^l(t-1) + Y_{[i_1, j_2], \mathbf{z}^r(t)}$. For the rest, the algorithm proceeds exactly as described in Section II.

Now, it is important to observe that marks of pairs on the RHS of the graph evolve exactly as the marks of a coupled PGM that operates over a pairs graph \mathcal{P}_R defined as follows. Denote the generic pair by $*_1, *_2$; then \mathcal{P}_R is a graph insisting on the set of nodes \mathcal{M}_r and in which the presence of edge $(\mathbf{z}^r(t), *_1, *_2)$, for any $*_1, *_2 \in \mathcal{P}_r \setminus \mathcal{Z}_t^r$, is dynamically unveiled at time t by observing variable $X_{z_1^l(t)*_1} X_{z_2^l(t)*_2} S_{z_1^l(t)*_1}^l S_{z_2^l(t)*_2}^r$. In other words, the edges originated from $\mathbf{z}^l(t)$ are replaced by the edges originated from $\mathbf{z}^r(t)$ and viceversa.

Furthermore, we make the following observations.

(i) We assume that the sequence of matched pairs $\{\mathbf{z}_t^R\}_t \in \mathcal{P}^{(R)}$ exactly corresponds to the sequence of matched pairs $\{\mathbf{z}^r(t)\}_t \in \mathcal{P}_r$, i.e., $\mathbf{z}^r(t) = \mathbf{z}^R(t)$ at every t . This is made possible by the fact that given $\mathcal{Z}_{t-1}^r = \mathcal{Z}_{t-1}^R$, marks collected by every unmatched pair in the two graphs at time t exactly correspond.

(ii) Our construction is consistent since edges between pairs are unveiled only once, specifically at the time at which the first between the two edge endpoints in \mathcal{P}_R is placed in $\mathcal{Z}_t^R = \mathcal{Z}_t^r$. Since then, the edge is replaced with an edge between two

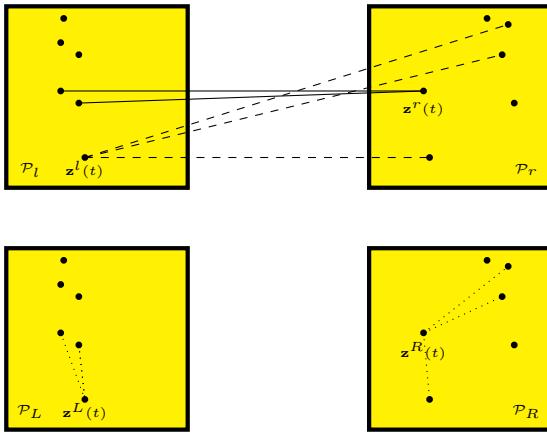


Fig. 11. Graphical representation of the PGM evolution over coupled graphs.

pairs that are both in \mathcal{P}_R , hence it will not be used again.

(iii) \mathcal{P}_R is isomorphic to a pairs graph originated by a generalized Erdős-Rényi graph \mathcal{G}_T^R , in which the presence of every edge $(z^r(t), *)$ can be represented by a Bernoulli r.v. and the probability that the edge is added to the graph takes values in the range $[p_{\min}, p_{\max}]$ and is independent of other edges. Indeed, observe that the presence of an edge in \mathcal{P}_R deterministically corresponds to the presence of the corresponding edge in $\mathcal{P}(\mathcal{G}_T)$. Furthermore, by construction, different edges in \mathcal{P}_R correspond to different edges in $\mathcal{P}(\mathcal{G}_T)$.

The same observations hold when we consider the evolution of the marks of the pairs on the left hand side and a pairs graph \mathcal{P}_L , which is originated from a coupled generalized Erdős-Rényi graph \mathcal{G}_T^L with same properties as \mathcal{G}_T^R .

Now, clearly $G(m, p_{\min}) \leq_{st} \mathcal{G}_T^R \leq_{st} G(m, p_{\max})$ and $G(m, p_{\min}) \leq_{st} \mathcal{G}_T^L \leq_{st} G(m, p_{\max})$, i.e., \mathcal{G}_T^R (\mathcal{G}_T^L) can be obtained by opportunely thinning a graph $G(m, p_{\max})$, while a graph $G(m, p_{\min})$ can be obtained by opportunely thinning \mathcal{G}_T^R (\mathcal{G}_T^L). Then we invoke Theorem 1 to conclude our proof and show that our algorithm correctly percolates over \mathcal{G}_T^R and \mathcal{G}_T^L and, thus, over the original bipartite \mathcal{G}_T .

Our matching procedure requires to extract (select) nodes that lie in a defined region \mathcal{H}_0 .

Clearly, to extract nodes lying in a defined region without errors, it is necessary to have direct access to vertices' positions. However, our algorithm has access only to graphs \mathcal{G}_1 and \mathcal{G}_2 (i.e., their adjacency matrix), and thus it extracts nodes based on "estimated" positions/distances (i.e. according to Theorem ?? or Proposition ??).

Thus, if we extract nodes on the basis of their estimated position, we will necessarily incur in some error: some nodes in \mathcal{H}_0 will not be selected while others lying outside \mathcal{H}_0 will be selected. We denote with $\mathcal{P}(\mathcal{H}_0)$ the set of pairs whose nodes lie in \mathcal{H}_0 and with $\hat{\mathcal{P}}(\mathcal{H}_0)$ the set of pairs composed by nodes that are extracted.

We need to devise a smart strategy that extracts nodes while guaranteeing that the following three conditions are satisfied:

- 1) Only good pairs formed by vertices whose actual location is in \mathcal{H}_0 (i.e. good pairs in $\mathcal{P}(\mathcal{H}_0)$) are extracted;
- 2) A finite fraction (bounded away from 0) of good pairs of

$\mathcal{P}(\mathcal{H}_0)$ is extracted (i.e., included in $\hat{\mathcal{P}}(\mathcal{H}_0)$);

- 3) The following situation occurs with negligible probability: a bad pair $[i_1, j_2]$ is included in $\hat{\mathcal{P}}(\mathcal{H}_0)$ while none of the pairs $[i_1, i_2]$ and $[j_1, j_2]$ are included.

The third condition ensures that every selected bad pair is in conflict with at least one good pair in the set, thus it will not be matched by the PGM algorithm when it (eventually) reaches the threshold. Below, we show how conditions 1) 2) and 3) can be easily guaranteed. For simplicity, we restrict our attention to spheric regions, although the same argument can be applied to regions of any shape.

We first introduce this preliminary result.

Proposition 4: Assume that position of nodes (length of edges) are estimated with a bounded error Δ . Then, given a spheric region \mathcal{H}_0 whose side is not smaller than 7Δ , it is possible to extract a set of nodes from \mathcal{G}_1 and \mathcal{G}_2 (and consequently to define $\hat{\mathcal{P}}(\mathcal{H}_0)$) satisfying conditions 1), 2) and 3).

Proof:

We select nodes as follows. We partition region \mathcal{H}_0 into three disjoint sub-regions. An inner spheric region of radius 3Δ co-centered within \mathcal{H}_0 , an intermediate annulus-shaped region with external radius equal to 5Δ , and a remaining outer region.

The idea is to extract only those pairs of vertices whose estimated position falls in either the inner or the intermediate region, under the additional condition that only pairs for which at least one vertex falls in the inner region are extracted. This expedient implies that $[i_1, j_2]$ is selected only if the estimated location of i_1 (j_2) falls in the inner region and the estimated position i_2 (j_1) falls in either the inner or the intermediate region. Clearly, the true position of i_1 (j_2) must necessarily lie in \mathcal{H}_0 . Furthermore, all nodes whose true position falls in a spheric region of radius Δ co-centered with \mathcal{H}_0 will be necessarily selected, thus conditions 1) and 2) are met w.h.p. as immediate consequence of Lemma 2. Finally, 3) is necessarily met as result of the following argument. (i) Observe that, for every node i , the distance between the estimated positions of i_1 and i_2 is by construction smaller than $2c_0C(n)$. (ii) Then let us consider a selected bad pair $[i_1, j_2]$; without lack of generality, we can assume the estimated position of i_1 to lie in the inner region. From consideration (i), the estimated position of i_2 must necessarily lie either in the inner or the intermediate region. (iii) As a result, the pair $[i_1, i_2]$ is necessarily selected too by our algorithm.

Proposition 5: The same approach can be pursued in the case of the application of Theorem 2 to define the initial set of vertices pairs $\mathcal{P}(\mathcal{N})$ so as to satisfy condition 3) (along with 1) and 2).

Indeed, in such a case the role of the inner region is played by $\mathcal{D}_{in}(\alpha_1\delta)$, the role of intermediate region is played by $\mathcal{D}_{in}(\alpha_2\delta) \setminus \mathcal{D}_{in}(\alpha_1\delta)$ while the role of outer region is played by $\mathcal{D}_{out}(\alpha_1\delta) \setminus \mathcal{D}_{in}(\alpha_1\delta)$. Indeed, by construction, if a vertex i_1 is accepted by adopting a threshold α_1 , the corresponding vertex i_2 will be necessarily accepted by adopting a threshold α_2 . ■

Lemma 2: The number $N_{\mathcal{H}_0}$ of nodes falling in a region \mathcal{H}_0 satisfies $\frac{n}{2}|\mathcal{H}_0| < N_{\mathcal{H}_0} < 2n|\mathcal{H}_0|$ w.h.p., as long as $|\mathcal{H}_0| =$

$\omega(\frac{1}{n})$. In particular, if $|\mathcal{H}_0| \geq c \frac{\log n}{n}$, then $\frac{n}{2} |\mathcal{H}_0| < N_{\mathcal{H}_0} < 2n |\mathcal{H}_0|$ with a probability $1 - O(n^{cH(1/2)})$.

Proof: The proof immediately descends by applying (7) and (8) to $N_{\mathcal{H}_0} = \text{Bin}(n, |\mathcal{H}_0|)$ with $\mu = \mathbb{E}[N_{\mathcal{H}_0}] = n|\mathcal{H}_0|$. ■

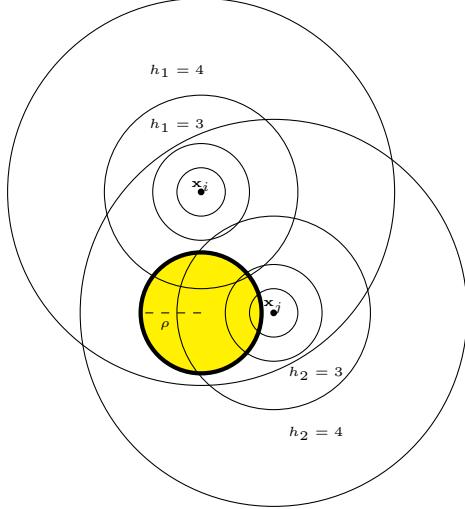


Fig. 12. Graphical representation of $\mathcal{D}_1(h_1)$ and $\mathcal{D}_2(h_2)$

Proof of Theorem 3. The proof of this proposition is based on the application of standard concentration results, namely, Chernoff bound and inequalities reported in Appendix A. For the sake of clarity, we restrict ourselves to consider the case $k = 2$; the extension to a generic k is easy to obtain.

Consider a correct pair $[i_1, i_2] \in \mathcal{D}(0, \rho_1) \setminus \mathcal{D}(0, \rho)$ whose location in \mathcal{H} is denoted by \mathbf{x}_i . We compute its number of edges with pairs in $\mathcal{D}(0, \rho)$, $N_i = \sum_{l \in \mathcal{D}(0, \rho)} Y[i_1, i_2][l_1 l_2] \geq \sum_{l \in \mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))} Y[i_1, i_2][l_1 l_2] = \text{Bin}(N_{\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))}, K(n))$ where $N_{\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))}$ denotes the number of nodes in $\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))$.

As an immediate consequence of Lemma 2, $N_{\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))} > \frac{n}{2} |\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))|$ with a probability $1 - O(n^{-2})$. Then, conditionally to this relation, we have $\text{Bin}(N_{\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))}, K(n)) > \frac{n}{2} |\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))| \frac{K(n)}{2}$ with a probability $1 - O(n^{-\gamma})$ with $\gamma > 0$, as it can be immediately shown by applying (7).

As a consequence, our algorithm successfully identifies almost all good pairs in $\mathcal{D}(0, \rho_1) \setminus \mathcal{D}(0, \rho)$ (i.e., $N_{\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))} - o(N_{\mathcal{D}(0, \rho) \cap \mathcal{D}(\mathbf{x}_i, C(n))})$) with a probability $1 - O(n^{-1})$, again, as a consequence of (7) when applied to the number of matched nodes in $\mathcal{D}(0, \rho)$.

Next, consider a bad pair $[i_1, j_2]$ whose nodes i and j are located respectively in \mathbf{x}_i and \mathbf{x}_j , with $|\mathbf{x}_i| = \rho_i$ and $|\mathbf{x}_j| = \rho_j$. Let $\mathcal{D}_1(h_1) = \mathcal{D}(\mathbf{x}_i, 2^{h_1+1}C(n)) \setminus \mathcal{D}(\mathbf{x}_i, 2^{h_1}C(n))$ for $h_1 \geq 1$ with $\mathcal{D}_1(0) = \mathcal{D}(\mathbf{x}_i, 2C(n))$ and $\mathcal{D}_2(h_2) = \mathcal{D}(\mathbf{x}_j, 2^{h_2+1}C(n)) \setminus \mathcal{D}(\mathbf{x}_j, 2^{h_2}C(n))$ with $\mathcal{D}_2(0) = \mathcal{D}(\mathbf{x}_j, 2C(n))$ (see Figure 12).

Let $\mathcal{C}(h_1, h_2) = \mathcal{D}(0, \rho) \cap \mathcal{D}_1(h_1) \cap \mathcal{D}_2(h_2)$ for $h_1 \geq 1$ and $h_2 \geq 0$. We have: $N_{[i_1, j_2]} = \sum_{l \in \mathcal{D}(0, \rho)} Y[i_1, j_2][l_1 l_2] = \sum_{h_1} \sum_{h_2} \sum_{l \in \mathcal{C}(h_1, h_2)} Y[i_1, j_2][l_1 l_2] \leq \sum_{h_1} \sum_{h_2} \text{Bin}(N_{\mathcal{C}(h_1, h_2)}, K^2(n) 2^{-\beta(h_1+h_2)})$.

Now, $\mathcal{C}(h_1, h_2)$ is a subset of both $\mathcal{C}(h_1) = \mathcal{D}(0, \rho) \cap \mathcal{D}_1(h_1)$ and $\mathcal{C}(h_2) = \mathcal{D}(0, \rho) \cap \mathcal{D}_2(h_2)$. Thus $N_{\mathcal{C}(h_1, h_2)} \leq \min(N_{\mathcal{C}(h_1)}, N_{\mathcal{C}(h_2)})$. In addition, by construction: $\mathcal{D}(0, \rho) \cap \mathcal{D}_1(h_1) = \emptyset$ if $h_1 < h_1^{\min} [\log_2(1 + \frac{\rho_i - \rho}{C(n)})]$, or $h_1^{\max} > [\log_2(1 + \frac{\rho_i + \rho}{C(n)})]$. Similarly, $\mathcal{D}(0, \rho) \cap \mathcal{D}_2(h_2) = \emptyset$ if $h_2 < h_2^{\min} [\log_2(1 + \frac{\rho_j - \rho}{C(n)})]$, or $h_2 > h_2^{\max} = [\log_2(1 + \frac{\rho_j + \rho}{C(n)})]$.

Hence,

$$N_{[i_1, j_2]} \leq \sum_{h_1^{\min}}^{h_1^{\max}} \sum_{h_2^{\min}}^{h_2^{\max}} \text{Bin}(\min(N_{\mathcal{C}(h_1)}, N_{\mathcal{C}(h_2)}), K^2(n) 2^{-\beta(h_1+h_2)}) \quad (12)$$

Now $C(h_1)$ is by construction a subset of $\mathcal{D}(0, \rho)$ as well as of $\mathcal{D}_1(h_1)$, thus $N_{C(h_1)} \leq \min(N_{\mathcal{D}(0, \rho)}, N_{\mathcal{D}_1(h_1)})$ and similarly $N_{C(h_2)} \leq \min(N_{\mathcal{D}(0, \rho)}, N_{\mathcal{D}_2(h_2)})$, thus:

$$N_{[i_1, j_2]} \leq \sum_{h_1^{\min}}^{h_1^{\max}} \sum_{h_2^{\min}}^{h_2^{\max}} \text{Bin}(\min(N_{\mathcal{D}(0, \rho)}, N_{\mathcal{D}_1(h_1)}, N_{\mathcal{D}_2(h_2)}), K^2(n) 2^{-\beta(h_1+h_2)}) \quad (13)$$

Note that $|\mathcal{D}(0, \rho)| = \pi \rho^2$ while $|\mathcal{D}_1(h_1)| \leq \pi 2^{2(h_1+1)} C^2(n)$, and, similarly, $|\mathcal{D}_2(h_2)| \leq \pi 2^{2(h_2+1)} C^2(n)$. As a consequence, since all these defined regions are larger than $C^2(n)$, from Lemma 2 we have that, uniformly on h_1 and h_2 , the number of nodes in these regions is not larger than $2n$ times the volume of the regions themselves. I.e., $N_{\mathcal{D}(0, \rho)} < 2n\pi\rho^2$, $N_{\mathcal{D}_1(h_1)} \geq 2n\pi 2^{2(h_1+1)} C^2(n)$ and $N_{\mathcal{D}_2(h_2)} < 2n\pi(2^{2((h_2+1))} C^2(n))$ with a probability $1 - O(n^2)$. Thus, by construction:

$$\begin{aligned} \mathbb{E}[N_{[i_1, j_2]}] &\leq \sum_{h_1^{\min}}^{h_1^{\max}} \sum_{h_2^{\min}}^{h_2^{\max}} \mathbb{E}[\text{Bin}(2nC^2(n) \\ &\min\left(2^{2(h_1+1)}, 2^{2(h_2+1)}, \pi \frac{\rho^2}{C^2(n)}\right), K^2(n) 2^{-\beta(h_1+h_2)})](1 - O(n^{-2})) \\ &\quad + nO(n^{-2}) \end{aligned} \quad (14)$$

Furthermore, $\min(2^{2(h_1+1)}, 2^{2(h_2+1)}) \leq 2^{2(h_1+h_2)+3}$. Then we can rewrite the previous expression as:

$$\begin{aligned} \mathbb{E}[N_{[i_1, j_2]}] &\leq \sum_{h_1^{\min}}^{h_1^{\max}} \sum_{h_2^{\min}}^{h_2^{\max}} \\ &\mathbb{E}[\text{Bin}(2nC^2(n) \min(2^{2(h_1+h_2)+3}, \\ &\pi \frac{\rho^2}{C^2(n)}), K^2(n) 2^{-\beta(h_1+h_2)})] + O(n^{-1}). \end{aligned} \quad (15)$$

Now, if $2^{2(h_1^{\min} + h_2^{\min})+3} < \pi \frac{\rho^2}{C^2(n)}$, we can bound:

$$\begin{aligned} \mathbb{E}[N_{[i_1, j_2]}] &\leq \\ &\sum_{h_1^{\max}} \sum_{h_2^{\max}} \mathbb{E}[\text{Bin}(n2^{2(h_1+h_2)+4}, K^2(n)2^{-\beta(h_1+h_2)})] + O(n^{-1}) \\ &nC^2(n) \sum_{h_1^{\min}} \sum_{h_2^{\min}} 2^{4+h_1(2-\beta)+h_2(2-\beta)} K^2(n) + O(n^{-1}) \leq \\ &nC^2(n)2^{4+(h_1^{\min}+h_2^{\max})(2-\beta)} K^2(n) \sum_0^\infty \sum_0^\infty 2^{h_1+h_2(2-\beta)} \\ &+ O(n^{-1}) = \\ &2nC^2(n)2^{3+(h_1^{\min}+h_2^{\min})(2-\beta)} K^2(n) \left(\frac{1}{1-2^{2-\beta}}\right)^2 + O(n^{-1}). \end{aligned} \quad (16)$$

If, instead, $2^{2(h_1^{\min} + h_2^{\min})+3} > \pi \frac{\rho^2}{C^2(n)}$, with similar arguments we can bound:

$$\mathbb{E}[N_{[i_1, j_2]}] \leq 2n\pi\rho^2 K^2(n)2^{-(h_1^{\min}+h_2^{\min})\beta} \left(\frac{1}{1-2^{-\beta}}\right)^2 + O(n^{-1})$$

Observe that, in general,

$$\mathbb{E}[N_{[i_1, j_2]}] = O(n[C(n)]^2 K^2(n))$$

with $\mathbb{E}[N_{[i_1, j_2]}] = \Theta(nC^2(n)K^2(n))$ only when $h_1^{\min} + h_2^{\min}$ is bounded. As a consequence, the bad pair $[i_1, j_2]$ will not reach threshold $r = \Theta(nC^2(n)K^2(n))$ w.h.p, as it can be immediately verified by applying Markov inequality. However, we need to show that jointly all bad pairs will remain below the threshold with a probability $1 - O(n^{-1})$. We can prove this stronger property first by deriving a tighter bound for the probability that a specific pair reaches the threshold, and then by applying the union bound on all pairs.

Considering again the bad pair $[i_1, j_2]$, $N_{[i_1, j_2]}$ can be rewritten as $N_{[i_1, j_2]} = \sum_{l \in \mathcal{H}, l \neq i, j} \mathbf{1}_{l \in \mathcal{D}(0, \rho)} Y[i_1, i_2][l_1 l_2]$, i.e., as a sum of independent Bernoulli random variables. Thus, we can apply Chernoff inequality to bound its tail. Recalling that by construction $r \gg \mathbb{E}[N_{[i_1, j_2]}]$, we have:

$$\begin{aligned} P(N_{[i_1, j_2]} \geq r) &\leq e^r \left(\frac{\mathbb{E}[N_{[i_1, j_2]}]}{r}\right)^r = \\ &e^{r(1-\log \frac{r}{\mathbb{E}[N_{[i_1, j_2]}]})}. \end{aligned} \quad (18)$$

From the definition of r , it follows that $r = cnC^2(n)K(n)$ with $c = \frac{|\mathcal{D}(0, \rho)| \cap |\mathcal{D}(p_1, C(n))|}{4C^2(n)} > 0$. Thus,

$$\begin{aligned} \log P(N_{[i_1, j_2]} \geq r) &\leq cnC^2(n)K(n) \\ &\left(1 - \log \frac{1}{K(n)} - (\beta - 2)(h_1^{\min} + h_2^{\min}) \log 2 + C_1\right) \end{aligned} \quad (19)$$

where C_1 is an opportune constant. By assumption, $nC^2(n)K(n) \geq \log n$ and $K(n) = o((\log n)^{-\gamma})$, hence

$$\log P(N_{[i_1, j_2]} \geq r) \leq -c \log n \cdot \omega(1).$$

Since for large n we have $c \log n \cdot \omega(1) > 3 \log n$, it turns out that every bad pair $[i_1, j_2]$, regardless the position of its vertices, reaches threshold r with a probability $O(n^{-3})$. By applying the union bound, we can claim that jointly all of such pairs will remain below the threshold r with a probability $O(n^{-1})$.

Algorithm 1 The PGM algorithm

- 1: $\mathcal{A}_0 = \mathcal{B}_0 = \mathcal{A}_0(n), \mathcal{Z}_0 = \emptyset$
 - 2: **while** $\mathcal{A}_t \setminus \mathcal{Z}_t \neq \emptyset$ **do**
 - 3: $t = t + 1$
 - 4: Randomly select a pair $[*, *] \in \mathcal{A}_{t-1} \setminus \mathcal{Z}_{t-1}$ and add one mark to all neighbor pairs of $[*, *]$ in $\mathcal{P}(\mathcal{G}_T)$.
 - 5: Let $\Delta\mathcal{B}_t$ be the set of all neighbor pairs of $[*, *]$ in $\mathcal{P}(\mathcal{G}_T)$ whose mark counter has reached threshold r at time t .
 - 6: Construct set $\Delta\mathcal{A}_t \subseteq \Delta\mathcal{B}_t$ as follows. Order the pairs in $\Delta\mathcal{B}_t$ in an arbitrary way, select them sequentially and test them for inclusion in $\Delta\mathcal{A}_t$
 - 7: **if** the selected pair in $\Delta\mathcal{B}_t$ has no conflicting pair in \mathcal{A}_{t-1} or $\Delta\mathcal{A}_t$ **then**
 - 8: Insert the pair in $\Delta\mathcal{A}_t$
 - 9: **else**
 - 10: Discard it
 - 11: $\mathcal{Z}_t = \mathcal{Z}_{t-1} \cup [*, *], \mathcal{B}_t = \mathcal{B}_{t-1} \cup \Delta\mathcal{B}_t, \mathcal{A}_t = \mathcal{A}_{t-1} \cup \Delta\mathcal{A}_t$
 - 12: **return** $T = t, \mathcal{Z}_T = \mathcal{A}_T$
-

The proof we propose complements the one provided in [8], which holds only under the assumption $p_{\min} \gg \sqrt{\frac{n^{-3/r-1}}{s^2}}$.

Here, we restrict to the case $p_{\min} = O\left(\sqrt{\frac{n^{-3/r-1}}{s^2}}\right)$. With reference to PGM algorithm reported in Figure 1, we define:

- $\mathcal{B}_t(\mathcal{G}_T)$ as the set of pairs in $\mathcal{P}(\mathcal{G}_T)$ that at time step t have already collected a least r marks. It is composed of good pairs $\mathcal{B}'_t(\mathcal{G}_T)$ and bad pairs $\mathcal{B}''_t(\mathcal{G}_T)$;
- $\mathcal{A}_t(\mathcal{G}_T)$ as the set of matchable pairs at time t . Similarly to $\mathcal{B}_t(\mathcal{G}_T)$, it comprises good pairs $\mathcal{A}'_t(\mathcal{G}_T)$ and bad pairs $\mathcal{A}''_t(\mathcal{G}_T)$. In general, $\mathcal{A}_t(\mathcal{G}_T)$ and $\mathcal{B}_t(\mathcal{G}_T)$ do not coincide as $\mathcal{B}_t(\mathcal{G}_T)$ may include conflicting pairs that are not present in $\mathcal{A}_t(\mathcal{G}_T)$;
- $\mathcal{Z}_t(\mathcal{G}_T)$ as the set of pairs that have been matched up to time t . By construction, $|\mathcal{Z}_t| = t, \forall t$.

Next, we define $T_{G_{p_{\min}}} = \min\{t \text{ s.t. } |\mathcal{A}_t(G(n, p_{\min}))| = t\}$ and $T_{G_{p_{\max}}} = \min\{t \text{ s.t. } |\mathcal{A}_t(G(n, p_{\max}))| = t\}$. By Theorem ??, we have that both $T_{G_{p_{\min}}}$ and $T_{G_{p_{\max}}}$ are equal to $n - o(n)$. Then inductively on $t, \forall t < \min(T_{G_{p_{\min}}}, T_{G_{p_{\max}}})$, w.h.p.:

$$|\mathcal{B}''_t(\mathcal{G}_T)| \leq |\mathcal{B}''_t((G(n, p_{\max})))| = \emptyset \quad (20)$$

In (20), the inequality descends by monotonicity of sets \mathcal{B}'_t with respect to “ \leq_{st} ”. The following equality descends from Corollary 1 in [8] applied to $G_{p_{\max}}$. We remark that, under our assumption on p_{\min} and p_{\max} , we have $t_0 = T$ in Corollary 1 in [8], along with:

$$\begin{aligned} |\mathcal{A}_t(\mathcal{G}_T)| &\stackrel{(a)}{=} |\mathcal{B}'_t(\mathcal{G}_T)| \stackrel{(b)}{\geq} \\ |\mathcal{B}'_t(G(n, p_{\min}))| &\stackrel{(c)}{=} |\mathcal{A}_t(G(n, p_{\min}))| \stackrel{(d)}{>} t. \end{aligned} \quad (21)$$

In (21), equality (a) is an immediate consequence of (20), inequality (b) holds by monotonicity of sets \mathcal{B}'_t with respect to “ \leq_{st} ”, while equality (c) descends from Theorem ???. Inequality (d) descends from the fact that we assume $t < T_{G_{p_{\min}}}$.

Thus, necessarily, $\mathcal{A}_T(\mathcal{G}_T) = T \geq \min(T_{G_{p_{\min}}}, T_{G_{p_{\max}}}) = n - o(n)$ and $\mathcal{B}''_T(\mathcal{G}_T) = \emptyset$.

REFERENCES

- [1] A. Narayanan, V. Shmatikov, “De-anonymizing social networks,” *IEEE Symposium on Security and Privacy*, 2009.
- [2] P. Pedarsani, D.-R. Figueiredo, M. Grossglauser, “A Bayesian method for matching two similar graphs without seeds,” *IEEE Allerton* 2013.
- [3] W. Peng, F. Li, X. Zou, J. Wu, “A two-stage deanonymization attack against anonymized social networks,” *IEEE Trans. on Computers*, 63(2), 2014.
- [4] N. Korula, S. Lattanzi, “An efficient reconciliation algorithm for social networks,” *PVLDB*, 2014.
- [5] P. Pedarsani, M. Grossglauser, “On the privacy of anonymized networks,” *SIGKDD*, 2011.
- [6] L. Yartseva, M. Grossglauser, “On the performance of percolation graph matching,” *COSN*, 2013.
- [7] S. Janson, T. Luczak, T. Turova, T. Vallier, “Bootstrap percolation on the random graph $G_{n,p}$,” *The Annals of Applied Probability*, 22(5), 2012.
- [8] C.F. Chiasserini, M. Garett, E.Leonardi, “De-anonymizing scale-free social networks by percolation graph matching,” *INFOCOM*, 2015.
- [9] K. Bringmann, T. Friedrich, A. Krohmer, “De-anonymization of heterogeneous random graphs in quasilinear time,” *22nd Annual European Symposium on Algorithms, ESA’14*.
- [10] M. Penrose, *Random Geometric Graphs*, Oxford University Press, 2003.
- [11] Add health public data set, wave I (online) <http://www.cpc.unc.edu/projects/addhealth>
- [12] L. Takac, M. Zabovsky, “Data analysis in public social networks,” *Int. Scientific Conf. and Int. Workshop Present Day Trends of Innovations*, 2012.
- [13] Pokec network dataset - KONECT, (website) <http://konect.uni-koblenz.de/networks/soc-pokec-relationships>